# IBM Power Systems
# Performance Capabilities Reference
# IBM i operating system Version 6.1

**January/April/October 2008**

This
document is intended for use by qualified performance related programmers or analysts from IBM, IBM Business Partners and IBM customers using the IBM Power™ Systems platform running IBM i operating system. Information in this document may be readily shared with IBM i customers to understand the performance and tuning factors in IBM i operating system 6.1 and earlier where applicable. **For the latest updates and for the latest on IBM i performance information, please refer to the Performance Management Website: http://www.ibm.com/systems/i/advantages/perfmgmt/index.html**

Requests for use of performance information by the technical trade press or consultants should be directed to Systems Performance Department V3T, IBM Rochester Lab, in Rochester, MN. 55901 USA.

**Note!**

Before using this information, be sure to read the general information under “Special Notices.”

**Twenty Fifth Edition (January/April/October 2008) SC41-0607-13**

This edition applies to IBM i operating System V6.1 running on IBM Power Systems.

You can request a copy of this document by download from IBM i Center via the System i Internet site at: http://www.ibm.com/systems/i/ . The Version 5 Release 1 and Version 4 Release 5 Performance Capabilities Guides are also available on the IBM iSeries Internet site in the "On Line Library", at: http://publib.boulder.ibm.com/pubs/html/as400/online/chgfrm.htm.

Documents are viewable/downloadable in Adobe Acrobat (.pdf) format. Approximately 1 to 2 MB download. Adobe Acrobat reader plug-in is available at: http://www.adobe.com .

To request the CISC version (V3R2 and earlier), enter the following command on VM:

**REQUEST V3R2 FROM FIELDSIT AT RCHVMW2 (your name)**

To request the IBM iSeries Advanced 36 version, enter the following command on VM:

**TOOLCAT MKTTOOLS GET AS4ADV36 PACKAGE**

# Table of Contents

# Special Notices

**DISCLAIMER NOTICE**

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. This information is presented along with general recommendations to assist the reader to have a better understanding of IBM(*) products. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

All performance data contained in this publication was obtained in the specific operating environment and under the conditions described within the document and is presented as an illustration. Performance obtained in other operating environments may vary and customers should conduct their own testing.

Information is provided "AS IS" without warranty of any kind.

The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.

Some information addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in IBM product announcements. The information is presented here to communicate IBM's current investment and development activities as a good faith effort to help with our customers' future planning.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

The following terms, which may or may not be denoted by an asterisk (*) in this publication, are trademarks of the IBM Corporation.

| | | |
|---|---|---|
| iSeries or AS/400 | System/370 | Operating System/400 |
| C/400 | IPDS | i5/OS |
| OS/400 | COBOL/400 | Application System/400 |
| System i5 | RPG/400 | OfficeVision |
| System i | CallPath | Facsimile Support/400 |
| PS/2 | DRDA | Distributed Relational Database Architecture |
| OS/2 | SQL/400 | Advanced Function Printing |
| DB2 | ImagePlus | Operational Assistant |
| AFP | VTAM | Client Series |
| IBM | APPN | Workstation Remote IPL/400 |
| SQL/DS | SystemView | Advanced Peer-to-Peer Networking |
| 400 | ValuePoint | OfficeVision/400 |
| CICS | DB2/400 | iSeries Advanced Application Architecture |
| S/370 | ADSM/400 | ADSTAR Distributed Storage Manager/400 |
| RPG IV | AnyNet/400 | IBM Network Station |
| AIX | | Lotus, Lotus Notes, Lotus Word Pro, Lotus 1-2-3 |
| Micro-partitioning | POWER4 | POWER4+ |
| POWER | POWER5 | POWER5+ |
| Power™ Systems | POWER6 | POWER6+ |
| PowerPC | Power™ Systems Software | Power™ Systems Software |

The following terms, which may or may not be denoted by a double asterisk (**) in this publication, are trademarks or registered trademarks of other companies as follows:

| | |
|---|---|
| TPC Benchmark | Transaction Processing Performance Council |
| TPC-A, TPC-B | Transaction Processing Performance Council |
| TPC-C, TPC-D | Transaction Processing Performance Council |
| ODBC, Windows NT Server, Access | Microsoft Corporation |
| Visual Basic, Visual C++ | Microsoft Corporation |
| Adobe PageMaker | Adobe Systems Incorporated |
| Borland Paradox | Borland International Incorporated |
| CorelDRAW! | Corel Corporation |
| Paradox | Borland International |
| WordPerfect | Satelite Software International |
| BEST/1 | BGS Systems, Inc. |
| NetWare | Novell |
| Compaq | Compaq Computer Corporation |
| Proliant | Compaq Computer Corporation |
| BAPCo | Business Application Performance Corporation |
| Harvard | Gaphics Software Publishing Corporation |
| HP-UX | Hewlett Packard Corporation |
| HP 9000 | Hewlett Packard Corporation |
| INTERSOLV | Intersolve, Inc. |
| Q+E | Intersolve, Inc. |
| Netware | Novell, Inc. |
| SPEC | Systems Performance Evaluation Cooperative |
| UNIX | UNIX Systems Laboratories |
| WordPerfect | WordPerfect Corporation |
| Powerbuilder | Powersoft Corporation |
| SQLWindows | Gupta Corporation |
| NetBench | Ziff-Davis Publishing Company |
| DEC Alpha | Digital Equipment Corporation |

Microsoft, Windows, Windows 95, Windows NT, Internet Explorer, Word, Excel, and Powerpoint, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel, Intel Inside (logos), MMX and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.
Linux is a trademark of Linus Torvalds in the United States, other countries, or both.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Other company, product or service names may be trademarks or service marks of others.

## Purpose of this Document

**The intent of this document is to help provide guidance in terms of IBM i operating system performance, capacity planning information, and tips to obtain optimal performance on IBM i operating system.** This document is typically updated with each new release or more often if needed. This October 2008 edition of the IBM i V6.1 Performance Capabilities Reference Guide is an update to the April 2008 edition to reflect new product functions announced on October 7, 2008.

This edition includes performance information on newly announced IBM Power Systems including Power 520 and Power 550, utilizing POWER6 processor technology. This document further includes information on IBM System i 570 using POWER6 processor technology, IBM i5/OS running on IBM BladeCenter JS22 using POWER6 processor technology, recent System i5 servers (model 515, 525, and 595) featuring new user-based licensing for the 515 and 525 models and a new 2.3GHz model 595, DB2 UDB for iSeries SQL Query Engine Support, Websphere Application Server including WAS V6.1 both with the Classic VM and the IBM Technology for Java (32-bit) VM, WebSphere Host Access Transformation Services (HATS) including the IBM WebFacing Deployment Tool with HATS Technology (WDHT), PHP - Zend Core for i, Java including Classic JVM (64-bit), IBM Technology for Java (32-bit), IBM Technology for Java (64-bit) and bytecode verification, Cryptography, Domino 7, Workplace Collaboration Services (WCS), RAID6 versus RAID5 disk comparisons, new internal storage adapters, Virtual Tape, and IPL Performance.

The wide variety of applications available makes it extremely difficult to describe a "typical" workload. The data in this document is the result of measuring or modeling certain application programs in very specific and unique configurations, and should not be used to predict specific performance for other applications. The performance of other applications can be predicted using a system sizing tool such as IBM Systems Workload Estimator (refer to Chapter 22 for more details on Workload Estimator).

# Chapter 1. Introduction

IBM System i and IBM System p platforms unified the value of their servers into a single, powerful lineup of servers based on industry leading POWER6 processor technology with support for IBM i operating system (formerly known as i5/OS), IBM AIX and Linux for Power.

Following along with this exciting unification are a number of naming changes to the formerly named i5/OS, now officially called IBM i operating system. Specifically, recent versions of the operating system are referred to by IBM i operating system 6.1 and IBM i operating system 5.4, formerly i5/OS V6R1 and i5/OS V5R4 respectively. Shortened forms of the new operating system name are IBM i 6.1, i 6.1, i V6.1 iV6R1, and sometimes simply 'i'. As always, references to legacy hardware and software will commonly use the naming conventions of the time.

The Power 520 Express Edition is the entry member of the Power Systems portfolio, supporting both IBM i 5.4 and IBM i 6.1. The System i 570 is enhanced to enable medium and large enterprises to grow and extend their IBM i business applications more affordably and with more granularity, while offering effective and scalable options for deploying Linux and AIX applications on the same secure, reliable system.

The IBM Power 570 running IBM i offers IBM's fastest POWER6 processors in 2 to 16-way configurations, plus an array of other technology advances. It is designed to deliver outstanding price/performance, mainframe-inspired reliability and availability features, flexible capacity upgrades, and innovative virtualization technologies. New 5.0GHz and 4.4GHz POWER6 processors use the very latest 64-bit IBM POWER processor technology. Each 2-way 570 processor card contains one two-core chip (two processors) and comes with 32 MB of L3 cache and 8 MB of L2 cache.

The CPW ratings for systems with POWER6 processors are approximately 70% higher than equivalent POWER5 systems and approximately 30% higher than equivalent POWER5+ systems. For some compute-intensive applications, the new System i 570 can deliver up to twice the performance of the original 570 with 1.65 GHz POWER5 processors.

The 515 and 525 models introduced in April 2007, introduce user-based licensing for IBM i. For assistance in determining the required number of user licenses, see http://www.ibm.com/systems/i/hardware/515 (model 515) or http://www.ibm.com/systems/i/hardware/525 (model 525). User-based licensing is not a replacement for system sizing; instead, user-based licensing enables appropriate user connectivity to the system. Application environments require different amounts of system resources per user. See Chapter 22 (IBM Systems Workload Estimator) for assistance in system sizing.

Customers who wish to remain with their existing hardware but want to move to IBM i 6.1 may find functional and performance improvements. IBM i 6.1 continues to help protect the customer's investment while providing more function and better price/performance over previous

versions. The primary public performance information web site is found at:
http://www.ibm.com/systems/i/advantages/perfmgmt/index.html

# Chapter 2. iSeries and AS/400 RISC Server Model Performance Behavior

## 2.1 Overview

iSeries and AS/400 servers are intended for use primarily in client/server or other non-interactive work environments such as batch, business intelligence, network computing etc. 5250-based interactive work can be run on these servers, but with limitations. With iSeries and AS/400 servers, interactive capacity can be increased with the purchase of additional interactive features. Interactive work is defined as any job doing 5250 display device I/O. This includes:

| | |
|---|---|
| All 5250 sessions | RUMBA/400 |
| Any green screen interface | Screen scrapers |
| Telnet or 5250 DSPT workstations | Interactive subsystem monitors |
| 5250/HTML workstation gateway | Twinax printer jobs |
| PC's using 5250 emulation | BSC 3270 emulation |
| Interactive program debugging | 5250 emulation |
| PC Support/400 work station function | |

Note that printer work that passes through twinax media is treated as interactive, even though there is no "user interface". This is true regardless of whether the printer is working in dedicated mode or is printing spool files from an out queue. Printer activity that is routed over a LAN through a PC print controller are not considered to be interactive.

This explanation is different than that found in previous versions of this document. Previous versions indicated that spooled work would not be considered to be interactive and were in error.

As of January 2003, 5250 On-line Transaction Processing (OLTP) replaces the term "interactive" when referencing interactive CPW or interactive capacity. Also new in 2003, when ordering a iSeries server, the customer must choose between a Standard Package and an Enterprise Package in most cases. The Standard Packages comes with zero 5250 CPW and 5250 OLTP workloads are not supported. However, the Standard Package does support a limited 5250 CPW for a system administrator to manage various aspects of the server. Multiple administrative jobs will quickly exceed this capability. The Enterprise Package does not have any limits relative to 5250 OLTP workloads. In other words, 100% of the server capacity is available for 5250 OLTP applications whenever you need it.

5250 OLTP applications can be run after running the WebFacing Tool of IBM WebSphere Development Studio for iSeries and will require no 5250 CPW if on V5R2 and using model 800, 810, 825, 870, or 890 hardware.

### 2.1.1 Interactive Indicators and Metrics

Prior to V4R5, there were no system metrics that would allow a customer to determine the overall interactive feature capacity utilization. It was difficult for the customer to determine how much of the total interactive capacity he was using and which jobs were consuming interactive capacity. This got much easier with the system enhancements made in V4R5 and V5R1.

Starting with V4R5, two new metrics were added to the data generated by Collection Services to report the system's interactive CPU utilization (ref file QAPMSYSCPU). The first metric (SCIFUS) is the

interactive utilization - an average for the interval. Since average utilization does not indicate potential problems associated with peak activity, a second metric (SCIFTE) reports the amount of interactive utilization that occurred above threshold. Also, interactive feature utilization was reported when printing a System Report generated from Collection Services data. In addition, Management Central now monitors interactive CPU relative to the system/partition capacity.

Also in V4R5, a new operator message, CPI1479, was introduced for when the system has consistently exceeded the purchased interactive capacity on the system. The message is not issued every time the capacity is reached, but it will be issued on an hourly basis if the system is consistently at or above the limit. In V5R2, this message may appear slightly more frequently for 8xx systems, even if there is no change in the workload. This is because the message event was changed from a point that was beyond the purchased capacity to the actual capacity for these systems in V5R2.

In V5R1, Collection Services was enhanced to mark all tasks that are counted against interactive capacity (ref file QAPMJOBMI, field JBSVIF set to '1'). It is possible to query this file to understand what tasks have contributed to the system's interactive utilization and the CPU utilized by all interactive tasks. Note: the system's interactive capacity utilization may not be equal to the utilization of all interactive tasks. Reasons for this are discussed in Section 2.10, *Managing Interactive Capacity.*

With the above enhancements, a customer can easily monitor the usage of interactive feature and decide when he is approaching the need for an interactive feature upgrade.

### 2.1.2 Disclaimer and Remaining Sections

The performance information and equations in this chapter represent ideal environments. This information is presented along with general recommendations to assist the reader to have a better understanding of the iSeries server models. Actual results may vary significantly.

This chapter is organized into the following sections:

- Server Model Behavior
- Server Model Differences
- Performance Highlights of New Model 7xx Servers
- Performance Highlights of Current Model 170 Servers
- Performance Highlights of Custom Server Models
- Additional Server Considerations
- Interactive Utilization
- Server Dynamic Tuning (SDT)
- Managing Interactive Capacity
- Migration from Traditional Models
- Migration from Server Models
- Dedicated Server for Domino (DSD) Performance Behavior

### 2.1.3 V5R3

Beginning with V5R3, the processing limitations associated with the Dedicated Server for Domino (DSD) models have been removed. Refer to section 2.13, "*Dedicated Server for Domino Performance Behavior*", for additional information.

### 2.1.4 V5R2 and V5R1

There were several new iSeries 8xx and 270 server model additions in V5R1 and the i890 in V5R2. However, with the exception of the DSD models, the underlying server behavior did not change from V4R5. All 27x and 8xx models, including the new i890 utilize the same server behavior algorithm that was announced with the first 8xx models supported by V4R5. For more details on these new models, please refer to *Appendix C, "CPW, CIW and MCU Values for iSeries*".

Five new iSeries DSD models were introduced with V5R1. In addition, V5R1 expanded the capability of the DSD models with enhanced support of Domino-complementary workloads such as Java Servlets and WebSphere Application Server. Please refer to Section 2.13, *Dedicated Server for Domino Performance Behavior*, for additional information.

---

## 2.2 Server Model Behavior

### 2.2.1 In V4R5 - V5R2

Beginning with V4R5, all 2xx, 8xx and SBx model servers utilize an enhanced server algorithm that manages the interactive CPU utilization. This enhanced server algorithm may provide significant user benefit. On prior models, when interactive users exceed the interactive CPW capacity of a system, additional CPU usage visible in one or more CFINT tasks, reduces system capacity for all users including client/server. New in V4R5, the system attempts to hold interactive CPU utilization below the threshold where CFINT CPU usage begins to increase. Only in cases where interactive demand exceeds the limitations of the interactive capacity for an extended time (for example: from long-running, CPU-intensive transactions), will overhead be visable via the CFINT tasks. Highlights of this new algorithm include the following:

- As interactive users exceed the installed interactive CPW capacity, the response times of those applications may significantly lengthen and the system will attempt to manage these interactive excesses below a level where CFINT CPU usage begins to increase. Generally, increased CFINT may still occur but only for transient periods of time. Therefore, there should be remaining system capacity available for non-interactive users of the system even though the interactive capacity has been exceeded. It is still a good practice to keep interactive system use below the system interactive CPW threshold to avoid long interactive response times.
- Client/server users should be able to utilize most of the remaining system capacity even though the interactive users have temporarily exceeded the maximum interactive CPW capacity.
- The iSeries Dedicated Server for Domino models behave similarly when the Non Domino CPW capacity has been exceeded (i.e. the system attempts to hold Non Domino CPW capacity below the threshold where CFINT overhead is normally activated). Thus, Domino users should be able to run in the remaining system capacity available.
- With the advent of the new server algorithm, there is not a concept known as the interactive knee or interactive cap. The system just attempts to manage the interactive CPU utilization to the level of the interactive CPW capacity.
- Dynamic priority adjustment (system value QDYNPTYADJ) will not have any effect managing the interactive workloads as they exceed the system interactive CPW capacity. On the other hand, it won't hurt to have it activated.

- The new server algorithm only applies to the new hardware available in V4R5 (2xx, 8xx and SBx models). The behavior of all other hardware, such as the 7xx models is unchanged (see section 2.2.3 Existing Model section for 7xx algorithm).

### 2.2.2 Choosing Between Similarly Rated Systems

Sometimes it is necessary to choose between two systems that have similar CPW values but different processor megahertz (MHz) values or L2 cache sizes. If your applications tend to be compute intensive such as Java, WebSphere, EJBs, and Domino, you may want to go with the faster MHz processors because you will generally get faster response times. However, if your response times are already sub-second, it is not likely that you will notice the response time improvements. If your applications tend to be L2 cache friendly such as many traditional commercial applications are, you may want to choose the system that has the larger L2 cache. In either case, you can use the IBM eServer Workload Estimator to help you select the correct system (see URL: http://www.ibm.com/iseries/support/estimator ) .

### 2.2.3 Existing Older Models

Server model behavior applies to:
- AS/400 Advanced Servers
- AS/400e servers
- AS/400e custom servers
- AS/400e model 150
- iSeries model 170
- iSeries model 7xx

Relative performance measurements are derived from commercial processing workload (CPW) on iSeries and AS/400. CPW is representative of commercial applications, particularly those that do significant database processing in conjunction with journaling and commitment control.

Traditional (non-server) AS/400 system models had a single CPW value which represented the maximum workload that can be applied to that model. This CPW value was applicable to either an interactive workload, a client/server workload, or a combination of the two.

Now there are two CPW values. The larger value represents the maximum workload the model could support if the workload were entirely client/server (i.e. no interactive components). This CPW value is for the processor feature of the system. The smaller CPW value represents the maximum workload the model could support if the workload were entirely interactive. For 7xx models this is the CPW value for the interactive feature of the system.

**The two CPW values are NOT additive - interactive processing will reduce the system's client/server processing capability.** When 100% of client/server CPW is being used, there is no CPU available for interactive workloads. When 100% of interactive capacity is being used, there is no CPU available for client/server workloads.

For model 170s announced in 9/98 and all subsequent systems, the published interactive CPW represents the point (the "knee of the curve") where the interactive utilization may cause increased overhead on the system. (As will be discussed later, this threshold point (or knee) is at a different value for previously announced server models). Up to the knee the server/batch capacity is equal to the processor capacity (CPW) minus the interactive workload. As interactive requirements grow beyond the knee, overhead

grows at a rate which can eventually eliminate server/batch capacity and limit additional interactive growth. **It is best for interactive workloads to execute below (less than) the knee of the curve.** (However, for those models having the knee at 1/3 of the total interactive capacity, satisfactory performance can be achieved.) The following graph illustrates these points.

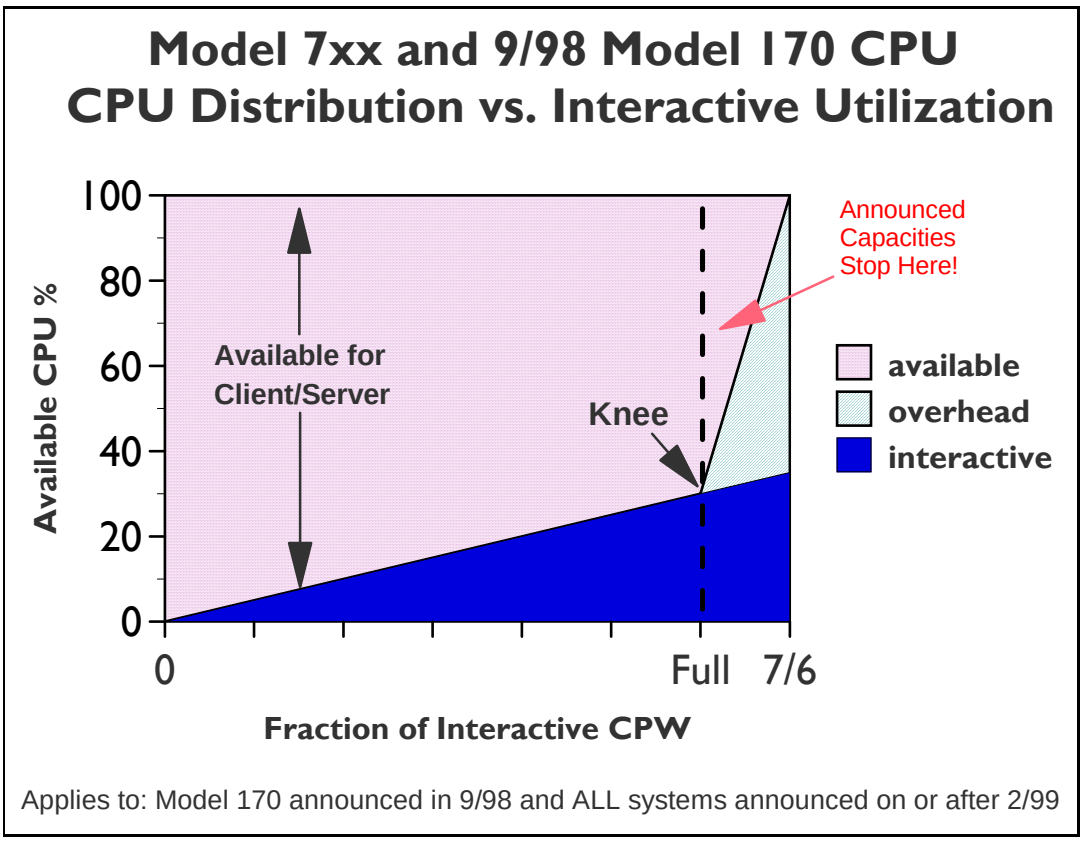



*Figure 2.1. Server Model behavior*

The figure above shows a straight line for the effective interactive utilization. Real/customer environments will produce a curved line since most environments will be dynamic, due to job initiation, interrupts, etc.

In general, a single interactive job will not cause a significant impact to client/server performance

Microcode task CFINTn, for all iSeries models, handles interrupts, task switching, and other similar system overhead functions. For the server models, when interactive processing exceeds a threshold amount, the additional overhead required will be manifest in the CFINTn task. Note that a single interactive job will not incur this overhead.

There is one CFINTn task for each processor. For example, on a single processor system only CFINT1 will appear. On an 8-way processor, system tasks CFINT1 through CFINT8 will appear. It is possible to see significant CFINT activity even when server/interactive overhead does not exist. For example if there are lots of synchronous or communication I/O or many jobs with many task switches.

The effective interactive utilization (EIU) for a server system can be defined as the useable interactive utilization plus the total of CFINT utilization.

## 2.3 Server Model Differences

Server models were designed for a client/server workload and to accommodate an interactive workload. When the interactive workload exceeds an interactive CPW threshold (the “knee of the curve”) the client/server processing performance of the system becomes increasingly impacted at an accelerating rate beyond the knee as interactive workload continues to build. Once the interactive workload reaches the maximum interactive CPW value, all the CPU cycles are being used and there is no capacity available for handling client/server tasks.

Custom server models interact with batch and interactive workloads similar to the server models but the degree of interaction and priority of workloads follows a different algorithm and hence the knee of the curve for workload interaction is at a different point which offers a much higher interactive workload capability compared to the standard server models.

For the server models the knee of the curve is approximately:

- 100% of interactive CPW for:
  - iSeries model 170s announced on or after 9/98
  - 7xx models
- 6/7 (86%) of interactive CPW for:
  - AS/400e custom servers
- 1/3 of interactive CPW for:
  - AS/400 Advanced Servers
  - AS/400e servers
  - AS/400e model 150
  - iSeries model 170s announced in 2/98

For the 7xx models the interactive capacity is a feature that can be sized and purchased like any other feature of the system (i.e. disk, memory, communication lines, etc.).

The following charts show the CPU distribution vs. interactive utilization for Custom Server and pre-2/99 Server models.

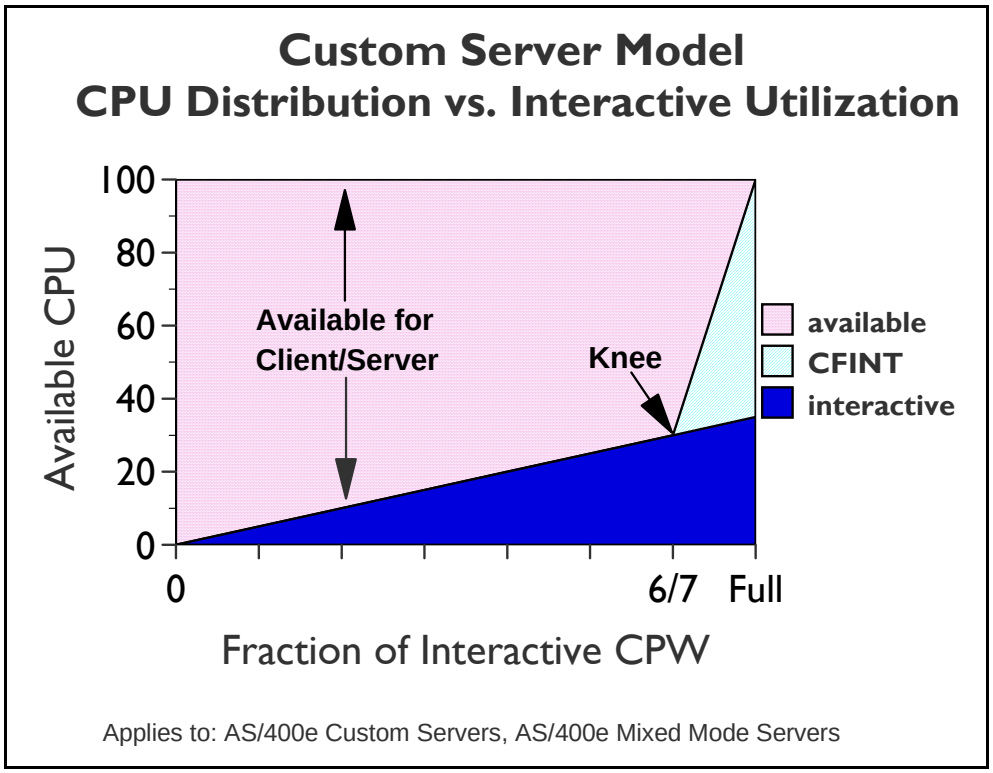



*Figure 2.2. Custom Server Model behavior*

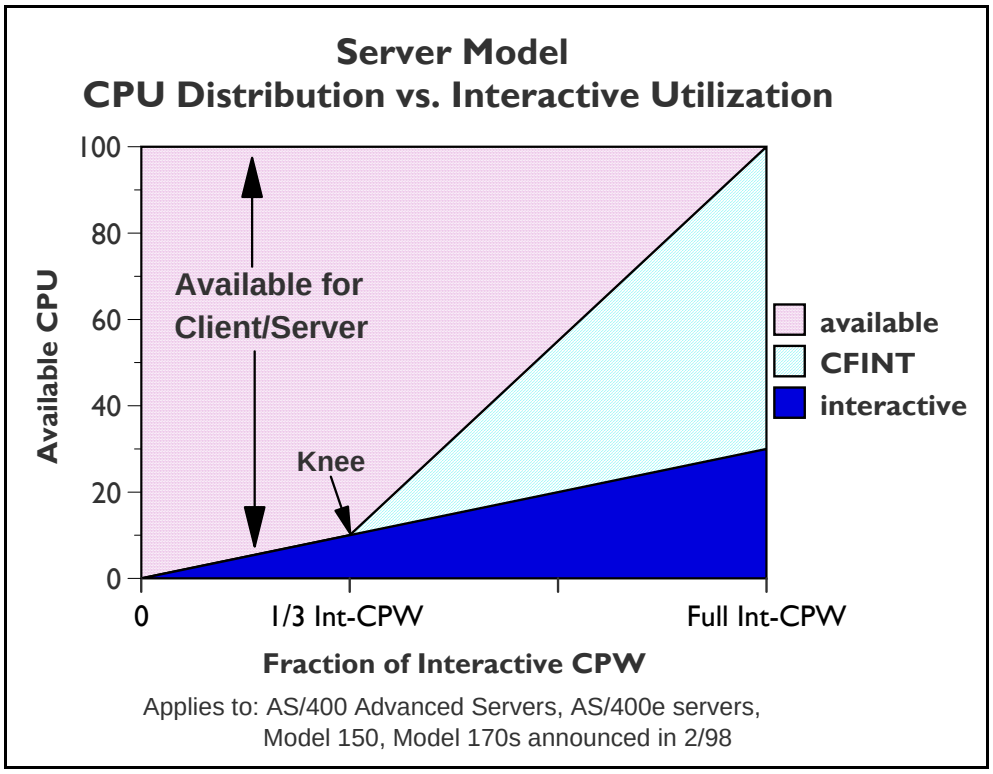



*Figure 2.3. Server Model behavior*

## 2.4 Performance Highlights of Model 7xx Servers

7xx models were designed to accommodate a mixture of traditional “green screen” applications and more intensive “server” environments. Interactive features may be upgraded if additional interactive capacity is required. This is similar to disk, memory, or other features.

Each system is rated with a **processor CPW** which represents the relative performance (maximum capacity) of a processor feature running a commercial processing workload (CPW) in a client/server environment. **Processor CPW** is achievable when the commercial workload is not constrained by main storage or DASD.

Each system may have one of several interactive features. Each interactive feature has an **interactive CPW** associated with it. **Interactive CPW** represents the relative performance available to perform host-centric (5250) workloads. The amount of interactive capacity consumed will reduce the available processor capacity by the same amount. The following example will illustrate this performance capacity interplay:

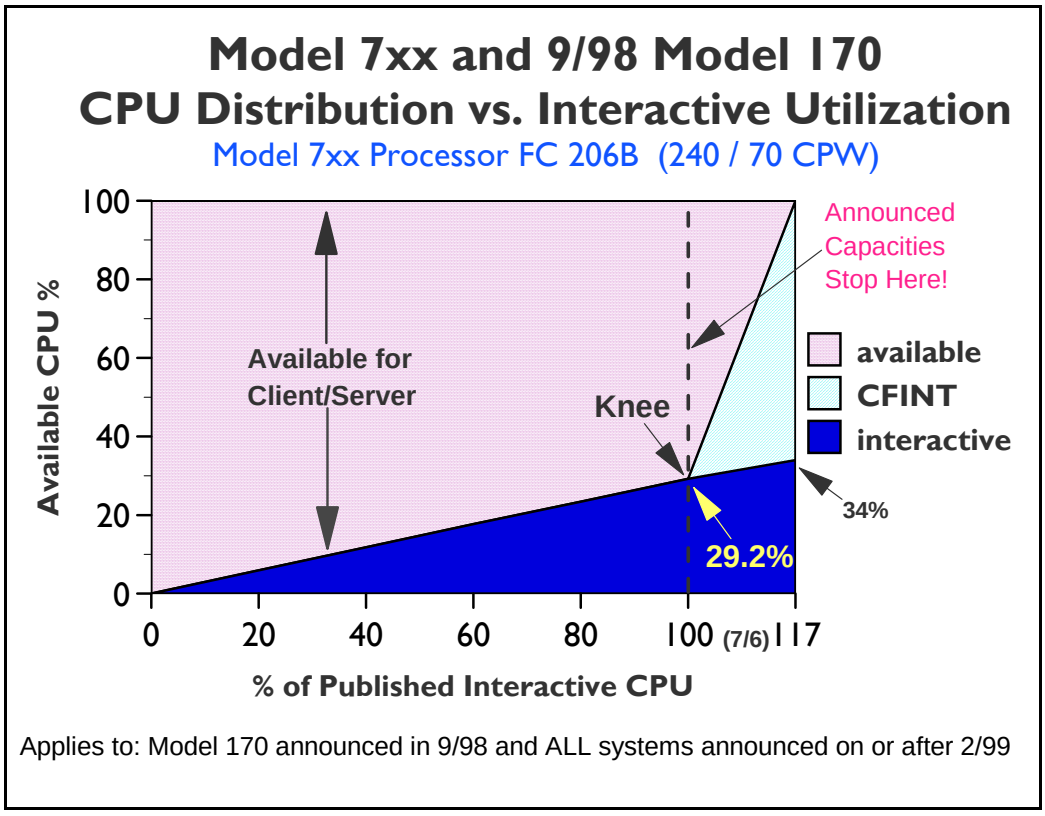



*Figure 2.4. Model 7xx Utilization Example*

At 110% of percent of the published interactive CPU, or 32.1% of total CPU, CFINT will use an additional 39.8% (approximate) of the total CPU, yielding an effective interactive CPU utilization of approximately 71.9%. This leaves approximately 28.1% of the total CPU available for client/server work. Note that the CPU is completely utilized once the interactive workload reaches about 34%. (CFINT would use approximately 66% CPU). At this saturation point, there is no CPU available for client/server.

## 2.5 Performance Highlights of Model 170 Servers

iSeries Dedicated Server for Domino models will be generally available on September 24, 1999. Please refer to Section 2.13, *iSeries Dedicated Server for Domino Performance Behavior*, for additional information.

Model 170 servers (features 2289, 2290, 2291, 2292, 2385, 2386 and 2388) are significantly more powerful than the previous Model 170s announced in Feb. '98. They have a faster processor (262MHz vs. 125MHz) and more main memory (up to 3.5GB vs. 1.0GB). In addition, the interactive workload balancing algorithm has been improved to provide a linear relationship between the client/server (batch) and published interactive workloads as measured by CPW.

The CPW rating for the maximum client/server workload now reflects the relative processor capacity rather than the "system capacity" and therefore there is no need to state a "constrained performance" CPW. This is because some workloads will be able to run at processor capacity if they are not DASD, memory, or otherwise limited.

Just like the model 7xx, the current model 170s have a **processor capacity** (CPW) value and an i**nteractive capacity** (CPW) value. These values behave in the same manner as described in the **Performance highlights of new model 7xx servers** section.

As interactive workload is added to the current model 170 servers, the remaining available client/server (batch) capacity available is calculated as: **CPW (C/S batch) = CPW(processor) - CPW(interactive)** This is valid up to the published interactive CPW rating. As long as the interactive CPW workload does not exceed the published interactive value, then interactive performance and client/server (batch) workloads will be both be optimized for best performance. **Bottom line, customers can use the entire interactive capacity with no impacts to client/server (batch) workload response times.**

On the current model 170s, if the **published interactive capacity** is exceeded, system overhead grows very quickly, and the client/server (batch) capacity is quickly reduced and becomes zero once the interactive workload reaches 7/6 of the published interactive CPW for that model.

The absolute limit for dedicated interactive capacity on the current models can be computed by multiplying the published interactive CPW rating by a factor of 7/6. The absolute limit for dedicated client/server (batch) is the published processor capacity value. This assumes that sufficient disk and memory as well as other system resources are available to fit the needs of the customer's programs, etc. Customer workloads that would require more than 10 disk arms for optimum performance should not be expected to give optimum performance on the model 170, as 10 disk access arms is the maximum configuration.
When the model 170 servers are running less than the published interactive workload, no Server Dynamic Tuning (SDT) is necessary to achieve balanced performance between interactive and client/server (batch) workloads. However, as with previous server models, a system value (QDYNPTYADJ - Server Dynamic Tuning ) is available to determine how the server will react to work requests when interactive workload exceeds the "knee". If the QDYNPTYADJ value is turned on, client/server work is favored over additional interactive work. If it is turned off, additional interactive work is allowed at the expense of low-priority client/server work. QDYNPTYADJ only affects the server when interactive requirements exceed the published interactive capacity rating. The shipped default value is for QDYNPTYADJ to be turned on.

The next chart shows the performance capacity of the current and previous Model 170 servers.

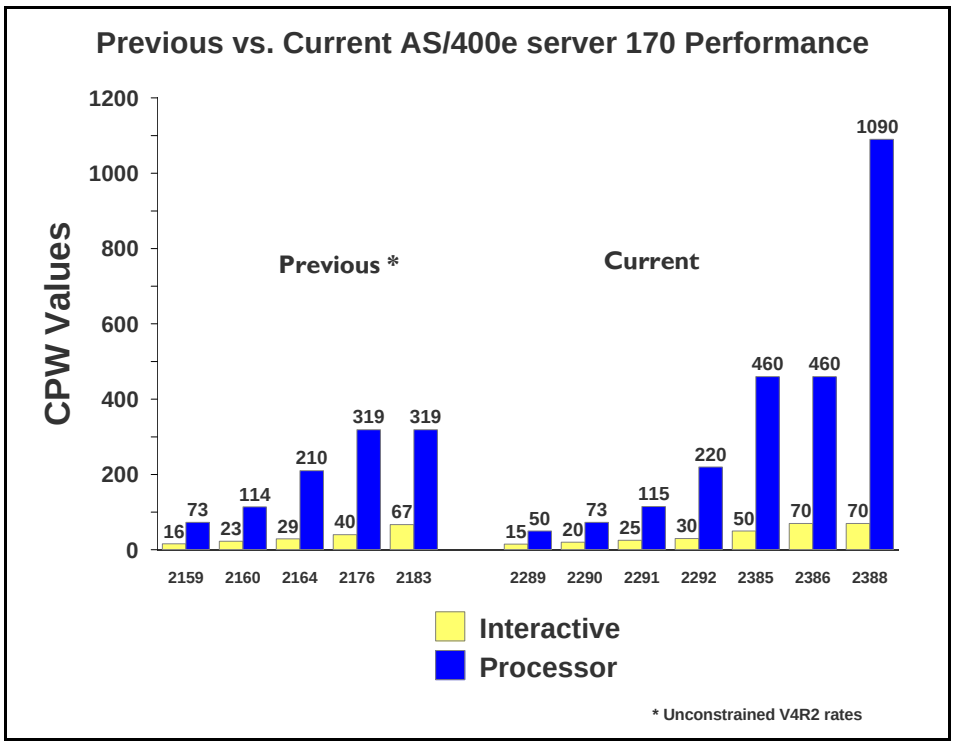



Figure 2.5. *Previous vs. Current Server 170 Performance*

## 2.6 Performance Highlights of Custom Server Models

Custom server models were available in releases V4R1 through V4R3. They interact with batch and interactive workloads similar to the server models but the degree of interaction and priority of workloads is different, and the knee of the curve for workload interaction is at a different point. When the interactive workload exceeds approximately 6/7 of the maximum interactive CPW (the knee of the curve), the client/server processing performance of the system becomes increasingly impacted. Once the interactive workload reaches the maximum interactive CPW value, all the CPU cycles are being used and there is no
capacity available for handling client/server tasks.

## 2.7 Additional Server Considerations

It is recommended that the System Operator job run at runpty(9) or less. This is because the possibility exists that runaway interactive jobs will force server/interactive overhead to their maximum. At this point it is difficult to initiate a new job and one would need to be able to work with jobs to hold or cancel runaway jobs.

You should monitor the interactive activity closely. To do this take advantage of PM/400 or else run Collection Services nearly continuously and query monitor data base each day for high interactive use

and higher than normal CFINT values. The goal is to avoid exceeding the threshold (knee of the curve) value of interactive capacity.

## 2.8 Interactive Utilization

When the interactive CPW utilization is beyond the knee of the curve, the following formulas can be used to determine the effective interactive utilization or the available/remaining client/server CPW. ***These equations apply to all server models*.**

**CPWcs(maximum) =** client/server CPW maximum value
**CPWint(maximum) =** interactive CPW maximum value
**CPWint(knee) =** interactive CPW at the knee of the curve
**CPWint =** interactive CPW of the workload

**X** is the ratio that says how far into the overhead zone the workload has extended:
**X = (CPWint - CPWint(knee)) / (CPWint(maximum) - CPWint(knee))**

**EIU =** Effective interactive utilization**.** In other words, the free running**, CPWint(knee)**, interactive plus the combination of interactive and overhead generated by **X.**
**EIU = CPWint(knee) + (X * (CPWcs(maximum) - CPWint(knee)))**

**CPW remaining for batch = CPWcs(maximum) - EIU**

**Example 1:**

A model 7xx server has a Processor CPW of **240** and an Interactive CPW of **70**.
The interactive CPU percent at the knee equals (70 CPW / 240 CPW) or **29.2**%.
The maximum interactive CPU percent (7/6 of the Interactive CPW ) equals (81.7 CPW / 240 CPW) or **34**%.

Now if the interactive CPU is held to less than **29.2**% CPU (the knee), then the CPU available for the System, Batch, and Client/Server work is **100% - the Interactive CPU used**.

If the interactive CPU is allowed to grow above the knee, say for example **32.1** % (110% of the knee), then the CPU percent remaining for the Batch and System is calculated using the formulas above:
**X = (32.1 - 29.2) / (34 - 29.2) = .604**
**EIU = 29.2 + (.604 * (100 - 29.2)) = 71.9%**

**CPW remaining for batch = 100 - 71.9 = 28.1%**

Note that a swing of + or - 1% interactive CPU yields a swing of effective interactive utilization **(EIU)** from 57% to 87%. Also note that on custom servers and 7xx models, environments that go beyond the interactive knee may experience erratic behavior.

**Example 2:**

A Server Model has a Client/Server CPW of **450** and an Interactive CPW of **50**.
The maximum interactive CPU percent equals (50 CPW / 450 CPW) or **11**%.
The interactive CPU percent at the knee is 1/3 the maximum interactive value. This would equal **4**%.

Now if the interactive CPU is held to less than **4**% CPU (the knee), then the CPU available for the System, Batch, and Client/Server work is **100% - the Interactive CPU used**.

If the interactive CPU is allowed to grow above the knee, say for example **9**% (or 41 CPW), then the CPU percent remaining for the Batch and System is calculated using the formulas above:

**X = (9 - 4) / (11 - 4) = .71** (percent into the overhead area)
**EIU = 4 + (.71 * (100 - 4)) = 72%**

**CPW remaining for batch = 100 - 72 = 28%**

Note that a swing of + or - 1% interactive CPU yields a swing of effective interactive utilization **(EIU)** from 58% to 86%.

On earlier server models, the dynamics of the interactive workload beyond the knee is not as abrupt, but because there is typically less relative interactive capacity the overhead can still cause inconsistency in response times.

---

## 2.9 Server Dynamic Tuning (SDT)

Logic was added in V4R1 and is still in use today so customers could better control the impact of interactive work on their client/server performance. Note that with the new Model 170 servers (features 2289, 2290, 2291, 2292, 2385, 2386 and 2388) this logic only affects the server when interactive requirements exceed the published interactive capacity rating. For further details see the section, **Performance highlights of current model 170 servers**.

Through dynamic prioritization, all interactive jobs will be put lower in the priority queue, approximately at the knee of the curve. Placing the interactive jobs at a lesser priority causes the interactive jobs to slow down, and more processing power to be allocated to the client/server processing. As the interactive jobs receive less processing time, their impact on client/server processing will be lessened. When the interactive jobs are no longer impacting client/server jobs, their priority will dynamically be raised again.

The dynamic prioritization acts as a regulator which can help reduce the impact to client/server processing when additional interactive workload is placed on the system. In most cases, this results in better overall throughput when operating in a mixed client/server and interactive environment, but it can cause a noticeable slowdown in interactive response.

To fully enable SDT, customers MUST use a non-interactive job run priority (RUNPTY parameter) value of 35 or less (which raises the priority, closer to the default priority of 20 for interactive jobs).

Changing the existing non-interactive job's run priority can be done either through the Change Job (CHGJOB) command or by changing the RUNPTY value of the Class Description object used by the non-interactive job. This includes IBM-supplied or application provided class descriptions.

Examples of IBM-supplied class descriptions with a run priority value higher than 35 include QBATCH and QSNADS and QSYSCLS50. Customers should consider changing the RUNPTY value for QBATCH and QSNADS class descriptions or changing subsystem routing entries to not use class descriptions QBATCH, QSNADS, or QSYSCLS50.

If customers modify an IBM-supplied class description, they are responsible for ensuring the priority value is 35 or less after each new release or cumulative PTF package has been installed. One way to do this is to include the Change Class (CHGCLS) command in the system Start Up program.

NOTE: Several IBM-supplied class descriptions already have RUNPTY values of 35 or less. In these cases no user action is required. One example of this is class description QPWFSERVER with RUNPTY(20). This class description is used by Client Access database server jobs QZDAINIT (APPC) and QZDASOINIT (TCP/IP).

The system deprioritizes jobs according to groups or "bands" of RUNPTY values. For example, 10-16 is band 1, 17-22 is band 2, 23-35 is band 3, and so on.

Interactive jobs with priorities 10-16 are an exception case with V4R1. Their priorities will not be adjusted by SDT. These jobs will always run at their specified 10-16 priority.

When only a single interactive job is running, it will not be dynamically reprioritized.

When the interactive workload exceeds the knee of the curve, the priority of all interactive jobs is decreased one priority band, as defined by the Dynamic Priority Scheduler, every 15 seconds. If needed, the priority will be decreased to the 52-89 band. Then, if/when the interactive CPW work load falls below the knee, each interactive job's priority will gradually be reset to its starting value when the job is dispatched.

If the priority of non-interactive jobs are not set to 35 or lower, SDT stills works, but its effectiveness is greatly reduced, resulting in server behavior more like V3R6 and V3R7. That is, once the knee is exceeded, interactive priority is automatically decreased. Assuming non-interactive is set at priority 50, interactive could eventually get decreased to the 52-89 priority band. At this point, the processor is slowed and interactive and non-interactive are running at about the same priority. (There is little priority difference between 47-51 band and the 52-89 band.) If the Dynamic Priority Scheduler is turned off, SDT is also turned off.

Note that even with SDT, the underlying server behavior is unchanged. Customers get no more CPU cycles for either interactive or non-interactive jobs. SDT simply tries to regulate interactive jobs once they exceed the knee of the curve.

Obviously systems can still easily exceed the knee and stay above it, by having a large number of interactive jobs, by setting the priority of interactive jobs in the 10-16 range, by having a small client/server workload with a modest interactive workload, etc. The entire server behavior is a partnership with customers to give non-interactive jobs the bulk of the CPU while not entirely shutting out interactive.

To enable the Server Dynamic Tuning enhancement ensure the following system values are on:
(the shipped defaults are that they are set on)

- QDYNPTYSCD - this improves the job scheduling based on job impact on the system.
- QDYNPTYADJ - this uses the scheduling tool to shift interactive priorities after the threshold is reached.

The Server Dynamic Tuning enhancement is most effective if the batch and client/server priorities are in the range of 20 to 35.

***Server Dynamic Tuning Recommendations***

On the new systems and mixed mode servers have the QDYNPTYSCD and QDYNPTYADJ system value set on. This preserves non-interactive capacities and the interactive response times will be dynamic beyond the knee regardless of the setting. Also set non-interactive class run priorities to less than 35.

On earlier servers and 2/98 model 170 systems use your interactive requirements to determine the settings. For "pure interactive" environments turn the QDYNPTYADJ system value off. in mixed environments with important non-interactive work, leave the values on and change the run priority of important non-interactive work to be less than 35.

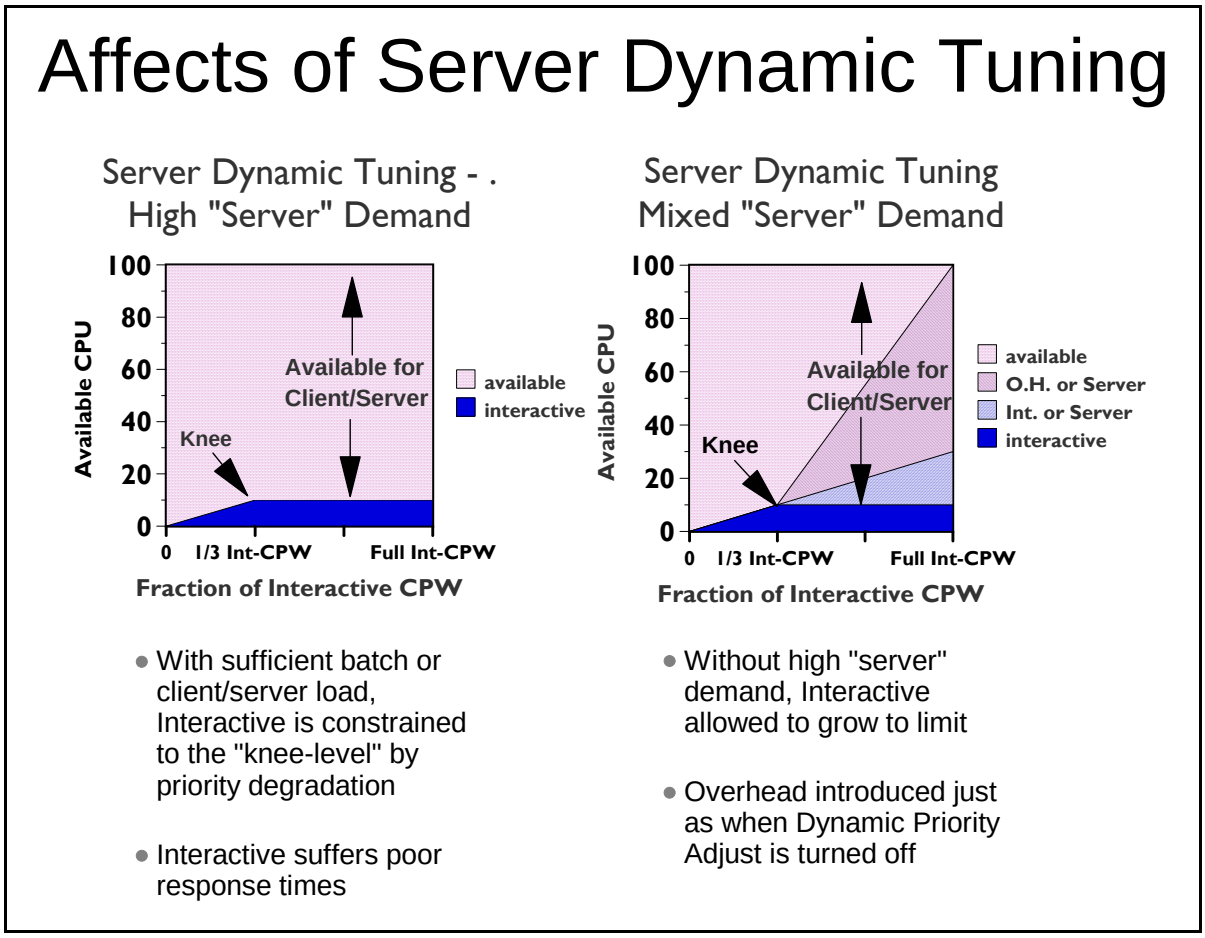



Figure 2.6.

## 2.10 Managing Interactive Capacity

***Interactive/Server characteristics in the real world.***

Graphs and formulas listed thus far work perfectly, provided the workload on the system is highly regular and steady in nature. Of course, very few systems have workloads like that. The more typical case is a dynamic combination of transaction types, user activity, and batch activity. There may very well be cases where the interactive activity exceeds the documented limits of the interactive capacity, yet decreases quickly enough so as not to seriously affect the response times for the rest of the workload. On the other hand, there may also be some intense transactions that force the interactive activity to exceed the documented limits interactive feature for a period of time even though the average CPU utilization appears to be less than these documented limits.

For 7xx systems, current 170 systems, and mixed-mode servers, a goal should be set to only rarely exceed the threshold value for interactive utilization. This will deliver the most consistent performance for both interactive and non-interactive work.

The questions that need to be answered are:
1. "How do I know whether my system is approaching the interactive limits or not?"
2. "What is viewed as 'interactive' by the system?"
3. "How close to the threshold can a system get without disrupting performance?"

This section attempts to answer these questions.

***Observing Interactive CPU utilization***

The most commonly available method for observing interactive utilization is Collection Services used in conjunction with the Performance Tools program product. The monitor collects system data as well as data for each job on the system, including the CPU consumed and the type of job. By examining the reports generated by the Performance Tools product, or by writing a query against the data in the various performance data base files.

Note: data is written to these files based on sample interval (Smallest is 5 minutes, default is 15 minutes). This data is an average for the duration of a measurement interval.

1. The first metric of interest is how much of the system's interactive capacity has been used. The file QAPMSYSCPU field SCIFUS contains the amount of interactive feature CPU time used. This metric became available with Collection Services in V4R5.

2. Even though average CPU may be reasonable your interactive workload may still be exceeding limits at times. The file QAPMSYSCPU field SCIFTE contains the amount of time the interactive threshold was exceeded during the interval. This metric became available with Collection Services in V4R5.

3. To determine what jobs are responsible for interactive feature consumption, you can look at the data in QAPMJOBL (Collection Services) or QAPMJOBS (Performance Monitor):
   - If using Collection Services on a V5R1 or later system, those jobs which the machine considers to be interactive are indicated by the field JBSVIF ='1'. These are all jobs that could contribute to your interactive feature utilization.
   - In all cases you can examine the jobs that are considered interactive by OS/400 as indicated by field JBTYPE = "I". Although not totally accurate, in most cases this will provide an adequate list of jobs that contributed to interactive utilization.

There are other means for determining interactive utilization. The easiest of these is the performance monitoring function of Management Central, which became available with V4R3. Management Central can provide:

- Graphical, real-time monitoring of interactive CPU utilization
- Creation of an alert threshold when an alert feature is turned on and the graph is highlighted
- Creation of an reverse threshold below which the highlights are turned off
- Multiple methods of handling the alert, from a console message to the execution of a command to the forwarding of the alert to another system.

By taking the ratio of the Interactive CPW rating and the Processor CPW rating for a system, one can determine at what CPU percentage the threshold is reached (This ratio works for the 7xx models and the current model 170 systems. For earlier models, refer to other sections of this document to determine what fraction of the Interactive CPW rating to use.) Depending on the workload, an alert can be set at some percentage of this level to send a warning that it may be time to redistribute the workload or to consider upgrading the interactive feature.

Finally, the functions of PM400 can also show the same type of data that Collection Services shows, with the advantage of maintaining a historical view, and the disadvantage of being only historical. However, signing up for the PM400 service can yield a benefit in determining the trends of how interactive capacities are used on the system and whether more capacity may be needed in the future.

***Is Interactive really Interactive?***

Earlier in this document, the types of jobs that are classified as interactive were listed. In general, these jobs all have the characteristic that they have a 5250 workstation communications path somewhere within the job. It may be a 5250 data stream that is translated into html, or sent to a PC for graphical display, but the work on the iSeries is fundamentally the same as if it were communicating with a real 5250-type display. However, there are cases where jobs of type "I" may be charged with a significant amount of work that is not "interactive". Some examples follow:

- Job initialization: If a substantial amount of processing is done by an interactive job's initial program, prior to actually sending and receiving a display screen as a part of the job, that processing may not be included as a part of the interactive work on the system. However, this may be somewhat rare, since most interactive jobs will not have long-running initial programs.

- More common will be parallel activities that are done on behalf of an interactive job but are not done within the job. There are two database-related activities where this may be the case.

    1. If the QQRYDEGREE system value is adjusted to allow for parallelism or the CHGQRYA command is used to adjust it for a single job, queries may be run in service jobs which are not interactive in nature, and which do not affect the total interactive utilization of the system. However, the work done by these service jobs is charged back to the interactive job. In this case, Collection Services and most other mechanisms will all show a higher sum of interactive CPU utilization than actually occurs. The exception to this is the WRKSYSACT command, which may show the current activity for the service jobs and/or the activity that they have "charged back" to the requesting jobs. Thus, in this situation it is possible for WRKSYSACT to show a lower system CPU utilization than the sum of the CPU consumption for all the jobs.

2. A similar effect can be found with index builds. If parallelism is enabled, index creation (CRTLF, Create Index, Open a file with MAINT(*REBUILD), or running a query that requires an index to be build) will be sent to service jobs that operate in non-interactive mode, but charge their work back to the job that requested the service. Again, the work does not count as "interactive", but the performance data will show the resource consumption as if they were.

- Lastly when only a single interactive job is running, the machine grants an exemption and does not include this job's activity in the interactive feature utilization.

There are two key ideas in the statements above. First, if the workload has a significant component that is related to queries or there is a single interactive job running, it will be possible to show an interactive job utilization in the performance tools that is significantly higher than what would be assumed and reported from the ratings of the Interactive Feature and the Processor Feature. Second, although it may make monitoring interactive utilization slightly more difficult, in the case where the workload has a significant query component, it may be beneficial to set the QQRYDEGREE system value to allow at least 2 processes, so that index builds and many queries can be run in non-interactive mode. Of course, if the nature of the query is such that it cannot be split into multiple tasks, the whole query is run inside the interactive job, regardless of how the system value is set.

***How close to the threshold can a system get without disrupting performance?***

The answer depends on the dynamics of the workload, the percentage of work that is in queries, and the projected growth rate. It also may depend on the number of processors and the overall capacity of the interactive feature installed. For example, a job that absorbs a substantial amount of interactive CPU on a uniprocessor may easily exceed the threshold, even though the "normal" work on the system is well under it. On the other hand, the same job on a 12-way can use at most 1/12th of the CPU, or 8.3%. a single, intense transaction may exceed the limit for a short duration on a small system without adverse affects, but on a larger system the chances of having multiple intense transactions may be greater.

With all these possibilities, how much of the Interactive feature can be used safely? A good starting point is to keep the average utilization below about 70% of the threshold value (Use double the threshold value for the servers and earlier Model 170 systems that use the 1/3 algorithm described earlier in this document.) If the measurement mechanism averages the utilization over a 15 minute or longer period, or if the workload has a lot of peaks and valleys, it might be worthwhile to choose a target that is lower than 70%. If the measurement mechanism is closer to real-time, such as with Management Central, and if the workload is relatively constant, it may be possible to safely go above this mark. Also, with large interactive features on fairly large processors, it may be possible to safely go to a higher point, because the introduction of workload dynamics will have a smaller effect on more powerful systems.

As with any capacity-related feature, the best answer will be to regularly monitor the activity on the system and watch for trends that may require an upgrade in the future. If the workload averages 60% of the interactive feature with almost no overhead, but when observed at 65% of the feature capacity it shows some limited amount of overhead, that is a clear indication that a feature upgrade may be required. This will be confirmed as the workload grows to a higher value, but the proof point will be in having the historical data to show the trend of the workload.

## 2.11 Migration from Traditional Models

This section describes a suggested methodology to determine which server model is appropriate to contain the interactive workload of a traditional model when a migration of a workload is occurring. It is assumed that the server model will have both interactive and client/server workloads.

To get the same performance and response time, from a CPU perspective, the interactive CPU utilization of the current traditional model must be known. Traditional CPU utilization can be determined in a number of ways. One way is to sum up the CPU utilization for interactive jobs shown on the Work with Active Jobs (WRKACTJOB) command.

```
**************************************************************************
                            Work with Active Jobs

 CPU %:   33.0     Elapsed time:   00:00:00     Active jobs:   152

 Type options, press Enter.

   2=Change   3=Hold   4=End   5=Work with   6=Release   7=Display message
   8=Work with spooled files   13=Disconnect ...

 Opt  Subsystem/Job  User        Type  CPU %  Function        Status
 __   BATCH          QSYS        SBS       0                 DEQW
 __   QCMN           QSYS        SBS       0                 DEQW
 __   QCTL           QSYS        SBS       0                 DEQW
 __     QSYSSCD      QPGMR       BCH       0  PGM-QEZSCNEP   EVTW
 __   QINTER         QSYS        SBS       0                 DEQW
 __     DSP05        TESTER      INT     0.2  PGM-BUPMENUNE  DSPW
 __     QPADEV0021   TEST01      INT     0.7  CMD-WRKACTJOB  RUN
 __   QSERVER        QSYS        SBS       0                 DEQW
 __     QPWFSERVSD   QUSER       BCH       0                 SELW
 __     QPWFSERVS0   QUSER       PJ        0                 DEQW
**************************************************************************
```

(Calculate the average of the CPU utilization for all job types "INT" for the desired time interval for interactive CPU utilization - "P" in the formula shown below.)

Another method is to run Collection Services during selected time periods and review the first page of the Performance Tools for iSeries licensed program Component Report. The following is an example of this section of the report:

**********

Component Report
Component Interval Activity
Data collected 190396 at 1030

Member . . . : Q960791030  Model/Serial . : 310-2043/10-0751D Main St...
Library. . : PFR  System name. . : TEST01  Version/Re..

| ITV End | Tns/hr | Rsp/Tns | CPU % Total | CPU% Inter | CPU % Batch | Disk I/O per sec Sync | Disk I/O per sec Async |
|---|---|---|---|---|---|---|---|
| 10:36 | 6,164 | 0.8 | 85.2 | 32.2 | 46.3 | 102.9 | 39 |
| 10:41 | 7,404 | 0.9 | 91.3 | 45.2 | 39.5 | 103.3 | 33.9 |
| 10:46 | 5,466 | 0.7 | 97.6 | 38.8 | 51 | 96.6 | 33.2 |
| 10:51 | 5,622 | 1.2 | 97.9 | 35.6 | 57.4 | 86.6 | 49 |
| 10:56 | 4,527 | 0.8 | 97.9 | 16.5 | 77.4 | 64.2 | 40.7 |
| : | | | | | | | |
| 11:51 | 5,068 | 1.8 | 99.9 | 74.2 | 25.7 | 56.5 | 19.9 |
| 11:56 | 5,991 | 2.4 | 99.9 | 46.8 | 45.5 | 65.5 | 32.6 |

Itv End------Interval end time (hour and minute)
Tns/hr-------Number of interactive transactions per hour
Rsp/Tns-----Average interactive transaction response time

**********

(Calculate the average of the CPU utilization under the "Inter" heading for the desired time interval for interactive CPU utilization - "P" in the formula shown below.)

It is possible to have interactive jobs that do not show up with type "INT" in Collection Services or the Component Report. An example is a job that is submitted as a batch job that acquires a work station. These jobs should be included in the interactive CPU utilization count.

Most systems have peak workload environments. Care must be taken to ensure that peaks can be contained in server model environments. **Some environments could have peak workloads that exceed the interactive capacity of a server model or could cause unacceptable response times and throughput.**

In the following equations, let the interactive CPU utilization of the existing traditional system be represented by percent P. A server model that should then produce the same response time and throughput would have a CPW of:

**Server Interactive CPW** = 3 * P * Traditional CPW

or for Custom Models use:

**Server Interactive CPW** = 1.0 * P * Traditional CPW (when P < 85%)

or

**Server interactive CPW** = 1.5 * P * Traditional CPW (when P >= 85%)

Use the 1.5 factor to ensure the custom server is sized less than 85% CPU utilization.

These equations provide the server interactive CPU cycles required to keep the interactive utilization at or below the knee of the curve, with the current interactive workload. The equations given at the end of the Server and Custom Server Model Behavior section can be used to determine the effective interactive utilization above the knee of the curve. The interactive workload below the knee of the curve represents

one third of the total possible interactive workload, for non-custom models. The equation shown in this section will migrate a traditional system to a server system and keep the interactive workload at or below the knee of the curve, that is, using less than two thirds of the total possible interactive workload. In some environments these equations will be too conservative. A value of 1.2, rather than 1.5 would be less conservative. The equations presented in the **Interactive Utilization** section should be used by those customers who understand how server models work above the knee of the curve and the ramifications of the V4R1 enhancement.

These equations are for migration of "existing workload" situations only. Installation workload projections for "initial installation" of new custom servers are generally sized by the business partner for 50 - 60% CPW workloads and no "formula increase" would be needed.

For example, assume a model 510-2143 with a single V3R6 CPW rating of 66.7 and assume the Performance Tools for iSeries report lists interactive work CPU utilization as 21%. Using the previous formula, the server model must have an interactive CPW rating of at least 42 to maintain the same performance as the 510-2143.

**Server interactive CPW = 3 * P * Traditional CPW**
**= 3 * .21 * 66.7**
**= 42**

A server model with an interactive CPW rating of at least 42 could approximate the same interactive work of the 510-2143, and still leave system capacity available for client/server activity. An S20-2165 is the first AS/400e series with an acceptable CPW rating (49.7).

Note that interactive and client/server CPWs are not additive. Interactive workloads which exceed (even briefly) the knee of the curve will consume a disproportionate share of the processing power and may result in insufficient system capacity for client/server activity and/or a significant increase in interactive response times.

## 2.12 Upgrade Considerations for Interactive Capacity

When upgrading a system to obtain more processor capacity, it is important to consider upgrading the interactive capacity, even if additional interactive work is not planned. Consider the following hypothetical example:

- The original system has a processor capacity of 1000 CPW and an interactive capacity of 250 ICPW
- The proposed upgrade system has a processor capacity of 4000 CPW and also offers an interactive capacity of 250 ICPW.
- On the original system, the interactive capacity allowed 25% of the total system to be used for interactive work. On the new system, the same interactive capacity only allows 6.25% of the total system to be used for interactive work.
- Even though the total interactive capacity of the system has not changed, the faster processors (and likely larger memory and faster disks) will allow interactive requests to complete more rapidly, which can cause greater spikes of interactive demand.
- So, just as it is important to consider balancing memory and disk upgrades with processor upgrades, optimal performance may also require an interactive capacity upgrade when moving to a new system.

## 2.13 iSeries for Domino and Dedicated Server for Domino Performance Behavior

In preparation for future Domino releases which will provides support for DB2 files, the previous processing limitations associated with DSD models have been removed in i5/OS V5R3.

In addition, a PTF is available for V5R2 which also removes the processing limitations for DSD models and allows full use of DB2. Please refer to PTF MF32968 and its prerequisite requirements.

The sections below from previous versions of this document are provided for those users on OS/400 releases prior to V5R3.

### 2.13.1 V5R2 iSeries for Domino & DSD Performance Behavior updates

Included in the V5R2 February 2003 iSeries models are five *iSeries for Domino* offerings. These include three i810 and two i825 models. The iSeries for Domino offerings are specially priced and configured for Domino workloads. There are no processing guidelines for the iSeries for Domino offerings as with non-Domino processing on the Dedicated Server for Domino models. With the iSeries for Domino offerings the full amount of DB2 processing is available, and it is no longer necessary to have Domino processing active for non-Domino applications to run well. Please refer to Chapter 11 for additional information on Domino performance in iSeries, and Appendix C for information on performance specifications for iSeries servers.

For existing iSeries servers, OS/400 V5R2 (both the June 2002 and the updated February 2003 version) will exhibit similar performance behavior as V5R1 on the Dedicated Server for Domino models. The following discussion of the V5R1 Domino-complimentary behavior is applicable to V5R2.

Five new DSD models were announced with V5R1. These included the iSeries Model 270 with a 1-way and a 2-way feature, and the iSeries Model 820 with 1-way, 2-way, and 4-way features. In addition, OS/400 V5R1 was enhanced to bolster DSD server capacity for robust Domino applications that require Java Servlet and WebSphere Application Server integration. The new behavior which supports Domino-complementary workloads on the DSD was available after September 28, 2001 with a refreshed version of OS/400 V5R1. This enhanced behavior is applicable to all DSD models including the model 170 and previous 270 and 820 models. Additional information on Lotus Domino for iSeries can be found in Chapter 11, “Domino for iSeries”.

For information on the performance behavior of DSD models for releases prior to V5R1, please refer the to V4R5 version of this document.

Please refer to Appendix C for performance specifications for DSD models, including the number of Mail and Calendaring Users (MCU) supported.

### 2.13.2 V5R1 DSD Performance Behavior

This section describes the performance behavior for all DSD models for the refreshed version of OS/400 V5R1 that was available after September 28, 2001.

A white paper, Enhanced V5R1 Processing Capability for the iSeries Dedicated Server for Domino, provides additional information on DSD behavior and can be accessed at: http://www.ibm.com/eserver/iseries/domino/pdf/dsdjavav5r1.pdf .

**Domino-Complementary Processing**
Prior to V5R1, processing that did not spend the majority of its time in Domino code was considered non-Domino processing and was limited to approximately 10-15% of the system capacity. With V5R1, many applications that would previously have been treated as non-Domino may now be considered as Domino-complementary when they are used in conjunction with Domino. Domino-complementary processing is treated the same as Domino processing, provided it also meets the criteria that the DB2 processing is less than 15% CPU utilization as described below. This behavioral change has been made to support the evolving complexity of Domino applications which frequently require integration with function such as Java Servlets and WebSphere Application Server. The DSD models will continue to have a zero interactive CPW rating which allows sufficient capacity for systems management processing. Please see the section below on Interactive Processing.

In other words, non-Domino workloads are considered complementary when used simultaneously with Domino, provided they meet the DB2 processing criteria. With V5R1, the amount of DB2 processing on a DSD must be less than 15% CPU. The DB2 utilization is tracked on a system-wide basis and all applications on the DSD cumulatively should not exceed 15% CPU utilization. Should the 15% DB2 processing level be reached, the jobs and/or threads that are currently accessing DB2 resources may experience increased response times. Other processing will not be impacted.

Several techniques can used to determine and monitor the amount of DB2 processing on DSD (and non-DSD) iSeries servers for V4R5 and V5R1.

- Work with System Status (WRKSYSSTS) command, via the ***% DB capability*** statistic
- Work with System Activity (WRKSYSACT) command which is part of the IBM Performance Tools for iSeries, via the ***Overall DB CPU util*** statistic
- Management Central - by starting a monitor to collect the ***CPU Utilization (Database Capability)*** metric
- Workload section in the System Report which can be generated using the IBM Performance Tools for iSeries, via the ***Total CPU Utilization (Database Capability)*** statistic

**V5R1 Non-Domino Processing**
Since all non-interactive processing is considered Domino-complementary when used simultaneously with Domino, provided it meets the DB2 criteria, non-Domino processing with V5R1 refers to the processing that is present on the system when there is no Domino processing present. (Interactive processing is a special case and is described in a separate section below). When there is no Domino processing present, all processing, including DB2 access, should be less than 10-15% of the system capacity. When the non-Domino processing capacity is reached, users may experience increased response times. In addition, CFINT processing may be present as the system attempts to manage the non-Domino processing to the available capacity. The announced “Processor CPW” for the DSD models refers to the amount of non-Domino processing that is supported .

Non-Domino processing on the 270 and 820 DSD models can be tracked using the Management Central function of Operations Navigator. Starting with V4R5, Management Central provides a special metric called “secondary utilization” which shows the amount of non-Domino processing. Even when Domino processing is present, the secondary utilization metric will include the Domino-complementary processing. And , as discussed above, the Domino-complementary processing running in conjunction with Domino will not be limited unless it exceeds the DB2 criteria.

**Interactive Processing**

Similar to previous DSD performance behavior for interactive processing, the Interactive CPW rating of 0 allows for system administrative functions to be performed by a single interactive user. In practice, a single interactive user will be able to perform necessary administrative functions without constraint. If multiple interactive users are simultaneously active on the DSD, the Interactive CPW capacity will likely be exceeded and the response times of those users may significantly lengthen. Even though the Interactive CPW capacity may be temporarily exceeded and the interactive users experience increased response times, other processing on the system will not be impacted. Interactive processing on the 270 and 820 DSD models can be tracked using the Management Central function of Operations Navigator.

**Logical Partitioning on a Dedicated Server**

With V5R1, iSeries logical partitioning is supported on both the Model 270 and Model 820. Just to be clear, iSeries logical partitioning is different from running multiple Domino partitions (servers). It is **not** necessary to use iSeries logical partitioning in order to be able to run multiple Domino servers on an iSeries system. iSeries logical partitioning lets you run multiple independent servers, each with its own processor, memory, and disk resources within a single symmetric multiprocessing iSeries. It also provides special capabilities such as having multiple versions of OS/400, multiple versions of Domino, different system names, languages, and time zone settings. For additional information on logical partitioning on the iSeries please refer to *Chapter 18. Logical Partitioning (LPAR)* and LPAR web at: http://www.ibm.com/eserver/iseries/lpar .

When you use logical partitioning with a Dedicated Server, the DSD CPU processing guidelines are pro-rated for each logical partition based on how you divide up the CPU capability. For example, suppose you use iSeries logical partitioning to create two logical partitions, and specify that each logical partition should receive 50% of the CPU resource. From a DSD perspective, each logical partition runs independently from the other, so you will need to have Domino-based processing in each logical partition in order for non-Domino work to be treated as complementary processing. Other DSD processing requirements such as the 15% DB2 processing guidelines and the 15% non-Domino processing guideline will be divided between the logical partitions based on how the CPU was allocated to the logical partitions. In our example above with 50% of the CPU in each logical partition, the DB2 database guideline will be 7.5% CPU for each logical partition. Keep in mind that WRKSYSSTS and other tools show utilization only for the logical partition they are running in; so in our example of a partition that has been allocated 50% of the processor resource, a 7.5% system-wide load will be shown as 15% within that logical partition. The non-Domino processing guideline would be divided in a similar manner as the DB2 database guideline.

**Running Linux on a Dedicated Server**

As with other iSeries servers, to run Linux on a DSD it is necessary to use logical partitioning. Because Linux is it's own unique operating environment and is not part of OS/400, Linux needs to have its own logical partition of system resources, separate from OS/400. The iSeries Hypervisor allows each partition to operate independently. When using logical partitioning on iSeries, the first logical partition, the primary partition, must be configured to run OS/400. For more information on running Linux on iSeries, please refer to *Chapter 13. iSeries Linux Performance* and Linux for iSeries web site at: Http://www.ibm.com/eserver/iseries/linux .

Running Linux in a DSD logical partition will exhibit different performance characteristics than running OS/400 in a DSD logical partition. As described in the section above, when running OS/400 in a DSD logical partition, the DSD capacities such as the 15% DB2 processing guideline and the 15% non-Domino processing guidelines are divided proportionately between the logical partitions based on how the processor resources were allocated to the logical partitions. However, for Linux logical partitions, the DSD guidelines are relaxed, and the Linux logical partition is able to use all of the resources allocated to it outside the normal guidelines for DSD processing. This means that it is not necessary to have Domino

processing present in the Linux logical partition, and all resources allocated to the Linux logical partition can essentially be used as though it were complementary processing. It is not necessary to proportionally increase the amount of Domino processing in the OS/400 logical partition to account for the fact that Domino processing is not present in the Linux logical partition .

By providing support for running Linux logical partitions on the Dedicated Server, it allows customers to run Linux-based applications, such as internet fire walls, to further enhance their Domino processing environment on iSeries. At the time of this publication, there is not a version of Domino that is supported for Linux logical partitions on iSeries.

# Chapter 3. Batch Performance

In a commercial environment, batch workloads tend to be I/O intensive rather than CPU intensive. The factors that affect batch throughput for a given batch application include the following:

- Memory (Pool size)
- CPU (processor speed)
- DASD (number and type)
- System tuning parameters

**Batch Workload Description**

The Batch Commercial Mix is a synthetic batch workload designed to represent multiple types of batch processing often associated with commercial data processing. The different variations allow testing of sequential vs random file access, changing the read to write ratio, generating "hot spots" in the data and running with expert cache on or off. It can also represent some jobs that run concurrently with interactive work where the work is submitted to batch because of a requirement for a large amount of disk I/O.

## 3.1 Effect of CPU Speed on Batch

The capacity available from the CPU affects the run time of batch applications. More capacity can be provided by either a CPU with a higher CPW value, or by having other contending applications on the same system consuming less CPU.

**Conclusions/Recommendations**

- For CPU-intensive batch applications, run time scales inversely with Relative Performance Rating (CPWs). This assumes that the number synchronous disk I/Os are only a small factor.

- For I/O-intensive batch applications, run time may not decrease with a faster CPU. This is because I/O subsystem time would make up the majority of the total run time.

- It is recommended that capacity planning for batch be done with tools that are available for iSeries. For example, PATROL for iSeries - Predict from BMC Software, Inc. * (PID# 5620FIF) can be used for modeling batch growth and throughput. BATCH400 (an IBM internal tool) can be used for estimating batch run-time.

## 3.2 Effect of DASD Type on Batch

For batch applications that are I/O-intensive, the overall batch performance is very dependent on the speed of the I/O subsystem. Depending on the application characteristics, batch performance (run time) will be improved by having DASD that has:

- faster average service times
- read ahead buffers
- write caches

Additional information on DASD devices in a batch environment can be found in Chapter 14, “DASD Performance”.

## 3.3 Tuning Parameters for Batch

There are several system parameters that affect batch performance. The magnitude of the effect for each of them depends on the specific application and overall system characteristics. Some general information is provided here.

- **Expert Cache**

  Expert Cache did not have a significant effect on the Commercial Mix batch workload. Expert Cache does not start to provide improvement unless the following are true for a given workload. These include:
  - the application that is running is disk intensive, and disk I/O's are limiting the throughput.
  - the processor is under-utilized, at less than 60%.
  - the system must have sufficient main storage.

  For Expert Cache to operate effectively, there must be spare CPU, so that when the average disk access time is reduced by caching in main storage, the CPU can process more work. In the Commercial Mix benchmark, the CPU was the limiting factor.

  However, specific batch environments that are DASD I/O intensive, and process data sequentially may realize significant performance gains by taking advantage of larger memory sizes available on the RISC models, particularly at the high-end. Even though in general applications require more main storage on the RISC models, batch applications that process data sequentially may only require slightly more main storage on RISC. Therefore, with larger memory sizes in conjunction with using Expert Cache, these applications may achieve significant performance gains by decreasing the number of DASD I/O operations.

- **Job Priority**

  Batch jobs can be given a priority value that will affect how much CPU processing time the job will get. For a system with high CPU utilization and a batch job with a low job priority, the batch throughput may be severely limited. Likewise, if the batch job has a high priority, the batch throughput may be high at the expense of interactive job performance.

- **Dynamic Priority Scheduling**

  See 19.2, “Dynamic Priority Scheduling” for details.

- **Application Techniques**

  The batch application can also be tuned for optimized performance. Some suggestions include:

  - Breaking the application into pieces and having multiple batch threads (jobs) operate concurrently. Since batch jobs are typically serialized by I/O, this will decrease the overall required batch window requirements.

  - Reduce the number of opens/closes, I/Os, etc. where possible.

  - If you have a considerable amount of main storage available, consider using the Set Object Access (SETOBJACC) command. This command pre-loads the complete database file, database index, or program into the assigned main storage pool if sufficient storage is available . The objective is to

improve performance by eliminating disk I/O operations.

- If communications lines are involved in the batch application, try to limit the number of communications I/Os by doing fewer (and perhaps larger) larger application sends and receives. Consider blocking data in the application. Try to place the application on the same system as the frequently accessed data.

# Chapter 4. DB2 for i5/OS Performance

This chapter provides a summary of the new performance features of DB2 for i5/OS on V6R1, V5R4 and V5R3, along with V5R2 highlights. Summaries of selected key topics on the performance of DB2 for i5/OS are provided. General information and some recommendations for improving performance are included along with links to the latest information on these topics. Also included is a section of performance references for DB2 for i5/OS.

## 4.1 New for i5/OS V6R1

In i5/OS V6R1 there are several performance enhancements to DB2 for i5/OS. The evolution of the SQL Query Engine (SQE), with this release, again supports more queries. Some of the new function supported may also have a sizable effect on performance, including derived key indexes, decimal floating-point data type, and select from insert. Lastly, modifications specifically to improve performance were made in several key areas, including optimization improvements to produce more efficient access plans, reducing full open and optimization time, and path length reduction of some basic, high use paths.

### i5/OS V6R1 SQE Query Coverage

The query dispatcher controls whether an SQL query will be routed to SQE or to the Classic Query Engine (CQE). SQL queries with the following attributes, which were routed to CQE in previous releases, may now be routed to SQE in i5/OS V6R1:

- NLSS/CCSID translation between columns
- User-defined table functions
- Sort sequence
- Lateral correlation
- UPPER/LOWER functions
- UTF8/16 Normalization support (NORMALIZE_DATA INI option of *YES)
- LIKE with UTF8/UTF16 data
- Character based substring and length for UTF8/UTF16 data

Also, in V6R1, the default value for the QAQQINI option IGNORE_DERIVED_INDEX has changed from *NO to *YES. The default behavior will now be to run supported queries through SQE even if there is a select/omit logical file index created over any of the tables in the query. In V6R1 many types of derived indexes are now supported by the SQE optimizer and usage of the QAQQINI option IGNORE_DERIVED_INDEX only applies to select/omit logical file indexes.

SQL queries with the attributes listed above will be processed by the SQE optimizer and engine in V6R1. Due to the robust SQE optimizer potentially choosing a better plan along with the more efficient query engine processing, there is the potential for better performance with these queries than was experienced in previous releases.

SQL queries which continue to be routed to CQE in i5/OS V6R1 have the following attributes:

- INSERT WITH VALUES statement or the target of an INSERT with subselect statement
- Logical files referenced in the FROM clause
- Tables with Read Triggers
- Read-only queries with more than 1000 dataspaces or updateable queries with more than 256 dataspaces.

- DB2 Multisystem tables

New function available in V6R1 whose use may affect SQL performance are derived key indexes, decimal floating point data type support, and the select from insert statement. A derived key index can have an expression in place of a column name that can use built-in functions, user defined functions, or some other valid expression. Additionally, you can use the SQL CREATE INDEX statement to create a sparse index using a WHERE condition.

The decimal floating-point data type has been implemented in V6R1. A decimal floating-point number is an IEEE 754R number with a decimal point. The position of the decimal point is stored in each decimal floating-point value. The maximum precision is 34 digits. The range of a decimal floating-point number is either 16 or 34 digits of precision, and an exponent range of $10^{-383}$ to $10^{384}$ or $10^{-6143}$ to $10^{6144}$ respectively. Use of the new decimal floating-point data type depends on whether you desire the new functionality. In general, more CPU is used to process data of this type versus decimal or floating-point data. The increased amount of processing time needed depends on the processor technology being used. Power6 hardware has special hardware support for processing decimal floating-point data, while Power5 does not. Power6 hardware enables much better performance for decimal floating-point processing. The CPU used to process this data depends on other factors also, including the application code, the functions used, and the data itself. As an example, for a specific set of queries run over a particular database, ranges for increased processing time for decimal floating-point data versus either decimal or floating point are shown in the chart below in Figure 4.1. The query attribute column shows the type of operations over the decimal floating-point columns in the queries.

| Query Attribute | POWER5 Processor | POWER6 Processor |
|---|---|---|
| Select | 0% to 15% | 0% to 15% |
| Arithmetic ( +, -, *, / ) | 15% improved to 400% | 35% improved to 45% |
| Functions ( AVG, MAX, MIN, SUM, CHAR, TRUN) | 15% improved to 1200% | 35% improved to 300% |
| Casts ( to/from int, decimal, float) | 40% improved to 600% | 35% improved to 500% |
| Inserts, Updates, and Create Index | 0% to 20% | 0% to 35% |

**Figure 4.1 Processing time degradation with decimal floating-point data versus decimal or float**

Given the additional processing time needed for decimal floating-point data, the recommendation is to use this data type only when the increased precision and rounding capabilities are needed. It is also recommended to avoid conversions to and from this data type, when possible. It should not normally be necessary to migrate existing packed or zoned decimal fields within a mature data base file to the new decimal floating point data type. Any decimal fields in the file will be converted to decimal float in host variables, as provided by the languages and APIs chosen. That will, in many cases, be a better performer overall (especially including existing code considerations) than a migration of the data field to a new format.

The ability to insert records into a table and then select from those inserted records in one statement, called Select From Insert, has been added to V6R1. Using a single SQL statement to insert and then retrieve the records can perform much better than doing an insert followed by a select statement. The chart below in figure 4.2 shows an example of the performance of a basic select from insert compared to the insert followed by select when inserting/selecting various number of records, from 1 to 1000. The data is for a particular database and SQL queries, and one specific hardware and software configuration running V6R1 i5/OS. The ratio of the clock times for these operations is shown. A ratio of less than 1 indicates that the select from insert ran faster than the insert followed by a select. Select from insert using NEW TABLE performs better than insert then select for all quantities of rows inserted. Select from insert using FINAL TABLE performs better in the one row case, but takes longer with more rows. This is due to the additional locking needed with FINAL TABLE to insure the rows are not modified until

the statement is complete. The implementation to invoke the locking causes a physical DASD write to the journal for each record, which causes journal waits. Journal caching on allows the journal writes to accumulate in memory and have one DASD write per multiple journal entries, greatly reducing the journal wait time. So select from insert statements with FINAL TABLE run much faster with journal caching on. Figure 4.2 shows that select from insert with FINAL TABLE and journal caching on ran faster than the insert followed by select for all but the 1000 row insert size.

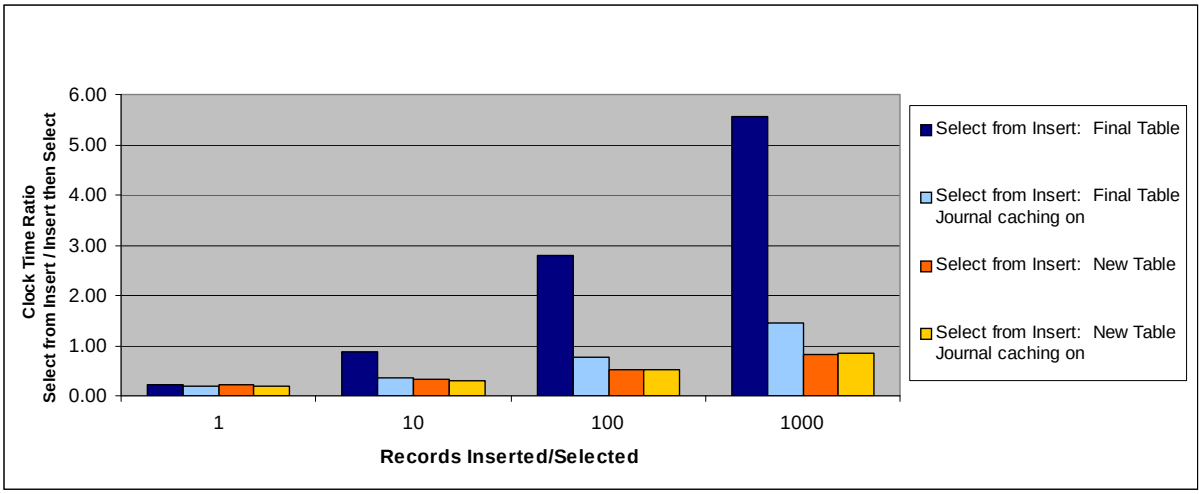



**Figure 4.2 Select from Insert versus Insert followed by Select clock time ratios**

In addition to updates for new functionality, in V6R1 substantial performance improvements were made to some SQL code paths. Improvements were made to the optimizer to make query execution cost estimates more accurate. This means that the optimizer is producing more efficient access plans for some queries, which may reduce their run time. The time required to full open and optimize queries was also largely reduced for many queries in V6R1. On average, for a group of greatly varying queries, the total open time including optimization has been reduced 45%. For a given set of very simple queries which go through a full open, but whose access plan already exists in the plan cache, the full open time was reduced by up to 30%.

In addition to the optimization and full open performance improvements, for V6R1 there was a comprehensive effort to reduce the basic path of a simple query which is running in re-use mode (pseudo open), and in particular is using JDBC to access the database. The results of this are potentially large reductions in the CPU time used in processing queries, particularly very simple queries. For a stock trade workload running through JDBC, throughput improvements of up to 78% have been measured. For more information please see Chapter 6. Web Server and WebSphere Performance.

## 4.2 DB2 i5/OS V5R4 Highlights

In i5/OS V5R4 there were several performance enhancements to DB2 for i5/OS. With support in SQE for Like/Substring, LOBs and the use of temporary indexes, many more queries now go down the SQE path. Thus there is the potential for better performance due to the robust SQE optimizer choosing a better plan along with the more efficient query engine processing. Also supported is use of Recursive Common

Table Expressions (RCTE) which allow for more elegant and better performing implementations of recursive processing. In addition, enhancements have been made in i5/OS V5R4 to the support for materialize query tables (MQTs) and partitioned table processing, which were both new in i5/OS V5R3.

**i5/OS V5R4 SQE Query Coverage**

The query dispatcher controls whether an SQL query will be routed to SQE or to CQE. SQL queries with the following attributes, which were routed to CQE in previous releases, may now be routed to SQE in i5/OS V5R4:

- Sensitive cursor
- Like/Substring predicates
- LOB columns
- ALWCPYDTA(*NO)

SQL queries which continue to be routed to CQE in i5/OS V5R4 have the following attributes:

- References to DDS logical files
- NLSS/CCSID translation between columns
- User-defined table unctions
- DB2 Multisystem
- Tables with select/omit logicals over them

In general, queries with Like and Substring predicates which are newly routed to SQE see substantial performance improvements in i5/OS V5R4. For a group of widely varying queries and data, including a wide range of Like and Substring predicates and various file sizes, a large percentage of the queries saw up to a 10X reduction in query run time. Queries with references to LOB columns, which were newly routed to SQE,, in general, also experience substantial performance improvements in i5/OS V5R4. For a set of queries which have references to LOB columns, in which the queries and data vary greatly a large percentage ran up to a 5X faster. .

A new addition to SQE is the creation and use of temporary indexes. These indexes will be created because they are required for implementing certain types of query requests or because they allow for better performance. The implementation of queries which require live data may require temporary indexes, for example, queries that run with a sensitive cursor or with ALWCPYDTA(*NO). In the case of using a temporary index for better performance, the SQE optimizer costs the creation and use of temporary indexes during plan optimization. An access plan will choose a temporary index if the amortized cost of building the index, provided one does not exist, reduces the query run time of the access plan enough that this plan wins over other plan options. The temporary indexes that the optimizer considers building are the same indexes in the 'index advised' list for a given query. Features unique to SQE temporary indexes, compared to CQE temporary indexes, are the longer lifetimes and higher degree of sharing of these indexes. SQE temporary indexes may be reused by the same query or other queries in the same job or in other jobs. The SQE temporary indexes will persist and will be maintained until the last query which references the temporary index is hard closed and the plan is removed from the plan cache. In many cases, this means the temporary indexes will persist until the last job which was using the index is ended. The high degree of sharing and longer lifetime allow for more reuse of the indexes without repeated create index cost.

New function for implementing applications that work with recursive data has been added to i5/OS V5R4. Recursive Common Table Expressions (RCTE) and Recursive Views may now be used in these types of applications, versus using SQL Stored Procedures and temporary results tables. For more information on using RCTEs and Recursive Views see the DB2 for System i *Database Performance and Query Optimization* manual.

Enhancements to extend the use of materialized query tables (MQTs) were added in i5/OS V5R4. New supported function in MQT queries by the MQT matching algorithm are unions and partitioned tables, along with limited support for scalar subselects, UDFs and user defined table functions, RCTE, and some scalar functions. Also new to i5/OS V5R4, the MQT matching algorithm now tries to match constants in the MQT with parameter markers or host variable values in the query. For more information on using MQTs see the DB2 for System i *Database Performance and Query Optimization* manual and the white paper, *The creation and use of materialized query tables within IBM DB2 FOR i5/OS*, available at http://www-304.ibm.com/jct09002c/partnerworld/wps/servlet/ContentHandler/SROY-6UZ5E6

The performance of queries which reference partitioned tables has been enhanced in i5/OS V5R4. The overhead when optimizing queries which reference a partitioned table has been reduced. Additionally, general improvements in plan quality have yielded run time improvements as well.

---

## 4.3 i5/OS V5R3 Highlights

In i5/OS V5R3, the SQL Query Engine (SQE) roll-out in DB2 for i5/OS took the next step. The new SQL Query Optimizer, SQL Query Engine and SQL Database Statistics were introduced in V5R2 with a limited set of queries being routed to SQE. In i5/OS V5R3 many more SQL queries are implemented in SQE. In addition, many performance enhancements were made to SQE in i5/OS V5R3 to decrease query runtime and to use System i resources more efficiently. Additional significant new features in this release are: table partitioning, the lookahead predicate generation (LPG) optimization technique for enhanced star-join support and a technology preview of materialized query tables. Also an April 2005 addition to the DB2 FOR i5/OS V5R3 support was query optimizer support for recognizing and using materialized query tables (MQTs) (also referred to as automatic summary tables or materialized views) for limited query functions. Two other improvements worth mentioning are faster delete support and SQE constraint awareness. This section contains a summary of the V5R3 information in the System i Performance Capabilities Reference i5/OS Version 5 Release 3 available at http://publib.boulder.ibm.com/infocenter/iseries/v5r3/topic/rzahx/sc410607.pdf.

### i5/OS V5R3 SQE Query Coverage

The query dispatcher controls whether an SQL query will be routed to SQE or to CQE (Classic Query Engine). The staged implementation of SQE enabled a very limited set of queries to be routed to SQE in V5R2. In general, read only single table queries with a limited set of attributes would be routed to SQE. The details of the query attributes for routing to SQE versus CQE in V5R2 are documented in the V5R2 redbook *Preparing for and Tuning the V5R2 SQL Query Engine*. With the V5R2 enabling PTF applied, PTF SI07650 documented in Info APAR II13486, the dispatcher routes many more queries through SQE. More single table queries and a limited set of multi-table queries are able to take advantage of the SQE enhancements. Queries with OR and IN predicates may be routed to SQE with the enabling PTF as will SQL queries with the appropriate attributes on systems with SMP enabled.

In i5/OS V5R3 a much larger set of queries are implemented in SQE including those with the enabling PTF on V5R2 and many queries with the following types of attributes:

- Subqueries
- Views
- Common table expressions
- Derived tables
- Unions
- Updates
- Deletes

SQL queries which continue to be routed to CQE in i5/OS V5R3 have the following attributes:

- Sensitive cursor
- Like/Substring predicates
- LOB columns
- References to DDS logical files
- NLSS/CCSID translation between columns
- DB2 Multisystem
- ALWCPYDTA(*NO)
- Tables with select/omit logicals over them

**i5/OS V5R3 SQE Performance Enhancements**

Many enhancements were made in i5/OS V5R3 to enable faster query runtime and use less system resource. Highlights of these enhancements include the following:

- New optimization techniques including Lookahead Predication Generation and Constraint Awareness
- Sharing of temporary result sets across jobs
- Reduction in size of temporary result sets
- More efficient I/O for temporary result sets
- Ability to do some aggregates with EVI symbol table access only
- Reduction in memory used during optimization
- Reduction in DB structure memory usage
- More efficient statistics generation during optimization
- Greater accuracy of statistics usage for optimization plan generation

The DB2 performance enhancements in i5/OS V5R3 substantially reduced the runtime of many queries. Performance improvements vary substantially due to many factors -- file size and layout, indexes and statistics available -- making generalization of performance expectations for a given query difficult. However, longer running queries which are newly routed to SQE in i5/OS V5R3, in general, have a greater likelihood of significant performance benefit.

For the short running queries, those that run less than 2 seconds, performance improvements are nominal. For subsecond queries there is little to no improvement for most queries. As the runtime increases, the reduction in runtime and CPU time become more substantial. In general, for short running queries there is less opportunity for improving performance. Also, the first execution of all the queries in these figures was measured so that a database open and full optimization were required. Database open and full optimization overhead may be higher with SQE, as it evaluates more information and examines more potential query implementation plans. As this overhead is much more expensive relative to actual query implementation for short running queries, performance benefits from SQE for the short running queries are minimized. However, in OLTP environments the plan caching and open data path (ODP) reuse design minimizes the number of opens and full optimizations needed. A very small percentage of queries in typical customer OLTP workloads go through full open and optimization.

The performance benefits are substantial for many of the medium to long running queries newly routed to SQE in i5/OS V5R3. Typically, the longer the runtime, the more potential for improvements. This is due to the optimizer constructing a more efficient access plan and the faster execution of the access plan with the SQE query engine. Many of the queries with runtimes greater than 2 seconds, especially those with runtimes greater than 10 seconds, reduced their runtime by a factor of 2 or more. Queries which run longer than 200 seconds were typically improved from 15% to over 100 times.

**Partitioned Table Support**

Table partitioning is a new feature introduced in i5/OS V5R3. The design is localized on an individual table basis rather than an entire library. The user specifies one or more fields which collectively act as a partitioning key. Next the records in the table are distributed into multiple disjoint sets based on the partitioning scheme used: either a system-supplied hashing function or a set of value ranges (such as dates by month or year) supplied by the user. The user can partition data using up to 256 partitions in i5/OS V5R3. The partitions are stored as multiple members associated with the same file object, which continues to represent the overall table as a single entity from an SQL data-access viewpoint.

The primary motivations for the initial release of this feature are twofold:

- Eliminate the limitation of at most 4 billion (2^32) rows in a single table
- Enhance data administration tasks such as save/restore, import/export, and add/drop which can be done more quickly on a partition basis (subset of a table)

In theory, table partitioning also offers opportunities for performance gains on queries that specify selection referencing a single or small number of partitions. In reality, however, the performance impact of partitioned tables in this initial release are limited on the positive side and may instead result in performance degradation when adopted too eagerly without carefully considering the ramifications of such a change. The resulting performance after partitioning a table depends critically on the mix of queries used to access the table and the number of partitions created. If fields used as partitioning keys are frequently included in selection criteria the resulting performance can be much better due to improved locality of reference for the desired records. When used incorrectly, table partitioning may degrade the performance of queries by an order of magnitude or more -- particularly when a large number of partitions (>32) are created.

Performance expectations of table partitioning on i5/OS V5R3 **should not** be equated at this time with partitioning concepts on other database platforms such as DB2 for Linux, Unix and Windows or offerings from other competitors. Nor should table partitioning on V5R3 be confused with the DB2 Multisystem for i5/OS offering. Carefully planned data storage schemes with active end-user disk arm management lead to the performance gains experienced with partitioned databases on those other platforms. Further gains are realized in other approaches through execution on clusters of physical nodes (in an approach similar to DB2 Multisystem for i5/OS). In addition, the entire schema is involved in the partitioning approach. On the other hand, the System i table partitioning design continues to utilize single level storage which already automatically spreads data to all disks in the relevant ASP. No new performance gains from I/O balancing are achieved when partitioning a table. Instead the gains tend to involve improved locality of reference for a subset of the data contained in a single partition or ease of administration when adding or deleting data on partition boundaries.

An in-depth discussion of table partitioning for i5/OS V5R3 is available in the white paper *Table Partitioning Strategies for DB2 FOR i5/OS* available at
http://www.ibm.com/servers/eserver/iseries/db2/awp.html

This publication covers additional details such as:

- Migration strategies for deployment
- Requirements and Limitations
- Sample Environments (OLTP, OLAP, Limits to Growth, etc.) & Recommended Settings
- Indexing Strategies

- Statistical Strategies
- SMP Considerations
- Administration Examples (Adding a Partition, Dropping a Partition, etc.)

**Materialized Query Table Support**

The initial release of i5/OS V5R3 includes the Materialized Query Table (MQT) (also referred to as automatic summary tables or materialized views) support in UDB DB2 for i5/OS as essentially a technology preview. Pre-April 2005 i5/OS V5R3 provides the capability of creating materialized query tables, but no optimizer awareness of these MQTs. An April 2005 addition to DB2 for i5/OS V5R3 is query optimizer support for recognizing and using MQTs. This additional support for recognizing and using MQTs is limited to certain query functions. MQTs can provide performance enhancements in a manner similar to indexes. This is done by precomputing and storing results of a query in the materialized query table. The database engine can use these results instead of recomputing them for a user specified query. The query optimizer will look for any applicable MQTs and can choose to implement the query using a given MQT provided this is a faster implementation choice. For long running queries, the run time may be substantially improved with judicious use of MQTs. For more information on MQTs including how to enable this new support, for which queries support MQTs and how to create and use MQTs see the DB2 for System i *Database Performance and Query Optimization* manual. For the latest information on MQTs see http://www-1.ibm.com/servers/eserver/iseries/db2/mqt.html.

**Fast Delete Support**

As developers have moved from native I/O to embedded SQL, they often wonder why a Clear Physical File Member (ClrPfm) command is faster than the SQL equivalent of DELETE FROM table. The reason is that the SQL DELETE statement deletes a single row at a time. In i5/OS V5R3, DB2 for System i has been enhanced with new techniques to speed up processing when every row in the table is deleted. If the DELETE statement is not run under commitment control, then DB2 for System i will actually use the ClrPfm operation underneath the covers. If the Delete is performed with commitment control, then DB2 FOR i5/OS can use a new method that's faster than the old delete one row at a time approach. Note however that not all DELETEs will use the new faster support. For example, delete triggers are still processed the old way.

---

## 4.4 V5R2 Highlights - Introduction of the SQL Query Engine

In V5R2 major enhancements, entitled SQL Query Engine (SQE), were implemented in DB2 for i5/OS. SQE encompasses changes made in the following areas:

- SQL query optimizer
- SQL query engine
- Database statistics

A subset of the read-only SQL queries are able to take advantage of these enhancements in V5R2.

**SQE Optimizer**

The SQL query optimizer has been enhanced with new optimization capabilities implemented in object oriented technology. This object oriented framework implements new optimization techniques and allows for future extendibility of the optimizer. Among the new capabilities of the optimizer are enhanced query access plan costing. For queries which can take advantage of the SQE enhancements,

more information may be used in the query plan costing phase than was available to the optimizer previously. The optimizer may now use newly implemented database statistics to make more accurate decisions when choosing the query access plan. Also, the enhanced optimizer may more often select plans using hash tables and sorted partial result lists to hold partial query results during query processing, rather than selecting access plans which build temporary indexes. With less reliance on temporary indexes the SQE optimizer is able to select more efficient plans which save the overhead of building temporary indexes and more fully take advantage of single-level store. The optimizer changes were designed to create efficient query access plans for the enhanced database engine.

**SQE Query Engine**

The database engine is the part of the database implementation which executes the access plan produced by the query optimizer. It accesses the data, processes it, and returns the SQL query results. The new engine enhancements, the SQE database engine, employ state of the art object oriented implementation. The SQE database engine was developed in tandem with the SQE optimization enhancements to allow for an efficient design which is readily extendable. Efficient new algorithms for the data access methods are used in query processing by the SQE engine.

The basic data access algorithms in SQE are designed to take full advantage of the System i single-level store to give the fastest query response time. The algorithms reduce I/O wait time by making use of available main memory and aggressively reading data from disk into memory. The goal of the data read-ahead algorithms is that the data is in memory when it is needed. This is done through the use of asynchronous I/Os. SQL queries which access large amounts of data may see a considerable improvement in the query runtime. This may also result in higher peak disk utilization.

The effects of the SQE enhancements on SQL query performance will vary greatly depending on many factors. Among these factors are hardware configuration (processor, memory size, DASD configuration...), system value settings, file layout, indexes available, query options file QAQQINI settings, and the SQL queries being run.

**SQE Database Statistics**

The third area of SQE enhancements is the collection and use of new database statistics. Efficient processing of database queries depends primarily on a query optimizer that is able to make judicious choices of access plans. The ability of an optimizer to make a good decision is critically influenced by the availability of database statistics on tables referenced in queries. In the past such statistics were automatically gathered during optimization time for columns of tables over which indexes exist. With SQE statistics on columns without indexes can now be gathered and will be used during optimization. Column statistics comprise histograms, frequent values list, and column cardinality.

With System i servers, the database statistics collection process is handled automatically, while on many platforms statistics collection is a manual process that is the responsibility of the database administrator. It is rarely necessary for the statistics to be manually updated, even though it is possible to manage statistics manually. The Statistics Manager determines on what columns statistics are needed, when the statistics collection should be run and when the statistics need to be refreshed. Statistics are automatically collected as low priority work in the background, so as to minimize the impact to other work on the system. The manual collection of statistics is run with the normal user job priority.

The system automatically determines what columns to collect statistics on based on what queries have run on the system. Therefore for queries which have slower than expected performance results, a check

should be made to determine if the needed statistics are available. Also in environments where long running queries are run only one time, it may be beneficial to ensure that statistics are available prior to running the queries.

Some properties of database column statistics are as follows:

- Column statistics occupy little storage, on average 8-12k per column.
- Column Statistics are gathered through one full scan of the database file for any given number of columns in the database file.
- Column statistics are maintained periodically through means of statistics refreshing mechanisms that require a full scan of the database file.
- Column statistics are packed in one concise data structure that requires few I/Os to page it into main memory during query optimization.

As stated above, statistics may have a direct effect on the quality of the access plan chosen by the query optimizer and thereby influence the end user query performance. Shown below is an illustrative example that underscores the effect of statistics on access plan selection process.

**Statistic Usage Example:**

Select * from T1, T2 where T1.A=T2.A and T1.B = 'VALUE1' and T2.C = 'VALUE2'

Database characteristics: indexes on T1.A and T2.A exist, NO column statistics, T1 has 100 million rows, T2 has 10 million rows. T1 is 1 GB and T2 0.1 GB

Since statistics are not available, the optimizer has to consider default estimates for selectivity of T1.B = 'VALUE1' ==> 10% T2.C = 'VALUE2' ==> 10%

The actual estimates are T1.B = 'VALUE1' ===>10% and T2.C = 'VALUE2' ===>0.00001%

Based on selectivity estimates the optimizer will select the following access plan

Scan(T1) - Probe (T2.A index) - > Probe (T2 Table) ---

the real cost for the above access plan would be approximately 8192 I/Os + 3600 I/Os ~ **11792 I/Os**

If column statistics existed on T2.C the selectivity estimate for T2.C = 'VALUE2' would be 10 rows or 0.00001%

And the query optimizer would select the following plan instead

Scan(T2) - Probe (T1.A index) - > Probe (T1 Table)

Accordingly the real cost could be calculated as follows:

819 I/Os + 10 I/Os ~ **830 I/Os. The result of having statistics on T2.C led to an access plan that is faster by order of magnitude from a case where no statistics exist .**

For more information on database statistics collection see the *DB2 for i5/OS Database Performance and Query Optimization* manual.

### SQE for V5R2 Summary

Enhancements to DB2 for i5/OS, called SQE, were made in V5R2. The SQE enhancements are object oriented implementations of the SQE optimizer, the SQE query engine and the SQE database statistics. In V5R2 a subset of the read-only SQL queries will be optimized and run with the SQE enhancements. The effect of SQE on performance will vary by workload and configuration. For the most recent information on SQE please see the SQE web page on the DB2 for i5/OS web site located at www.iseries.ibm.com/db2/sqe.html. More information on SQE for V5R2 is also available in the V5R2 redbook *Preparing for and Tuning the V5R2 SQL Query Engine*.

---

## 4.5 Indexing

Index usage can dramatically improve the performance of DB2 SQL queries. For detailed information on using indexes see the white paper *Indexing Strategies for DB2 for i5/OS* at http://www-1.ibm.com/servers/enable/site/education/abstracts/indxng_abs.html . The paper provides basic information about indexes in DB2 for i5/OS, the data structures underlying them, how the system uses them and index strategies. Also discussed are the additional indexing considerations related to maintenance, tools and methods.

### Encoded Vector Indices (EVIs)

DB2 for i5/OS supports the Encoded Vector Index (EVI) which can be created through SQL. EVIs cannot be used to order records, but in many cases, they can improve query performance. An EVI has several advantages over a traditional binary radix tree index.

- The query optimizer can scan EVIs and automatically build dynamic (on-the-fly) bitmaps much more quickly than from traditional indexes.

- EVIs can be built much faster and are much smaller than traditional indexes. Smaller indexes require less DASD space and also less main storage when the query is run.

- EVIs automatically maintain exact statistics about the distribution of key values, whereas traditional indexes only maintain estimated statistics. These EVI statistics are not only more accurate, but also can be accessed more quickly by the query optimizer.

EVIs are used by the i5/OS query optimizer with dynamic bitmaps and are particularly useful for advanced query processing. EVIs will have the biggest impact on the complex query workloads found in business intelligence solutions and ad-hoc query environments. Such queries often involve selecting a limited number of rows based on the key value being among a set of specific values (e.g. a set of state names).

When an EVI is created and maintained, a symbol table records each distinct key value and also a corresponding unique binary value (the binary value will be 1, 2, or 4 bytes long, depending on the number of distinct key values) that is used in the main part of the EVI, the vector (array). The subscript of each vector (array) element represents the relative record number of a database table row. The vector has an entry for each row. The entry in each element of the vector contains the unique binary value corresponding to the key value found in the database table row.

### 4.6 DB2 Symmetric Multiprocessing feature

#### Introduction

The DB2 SMP feature provides application transparent support for parallel query operations on a single tightly-coupled multiprocessor System i (shared memory and disk). In addition, the symmetric multiprocessing (SMP) feature provides additional query optimization algorithms for retrieving data. The database manager can automatically activate parallel query processing in order to engage one or more system processors to work simultaneously on a single query. The response time can be dramatically improved when a processor bound query is executed in parallel on multiple processors. For more information on access methods which use the SMP feature and how to enable SMP see the *DB2 for i5/OS Database Performance and Query Optimization* manual in the System i information center.

#### Decision Support Queries

The SMP feature is most useful when running decision support (DSS) queries. DSS queries which generally give answers to critical business questions tend to have the following characteristics:

- examine large volumes of data
- are far more complex than most OLTP transactions
- are highly CPU intensive
- includes multiple order joins, summarizations and groupings

DSS queries tend to be long running and can utilize much of the system resources such as processor capacity (CPU) and disk. For example, it is not unusual for DSS queries to have a response time longer than 20 seconds. In fact, complex DSS queries may run an hour or longer. The CPU required to run a DSS query can easily be 100 times greater than the CPU required for a typical OLTP transaction. Thus, it is very important to choose the right System i for your DSS query and data warehousing needs.

#### SMP Performance Summary

The SMP feature provides performance improvement for query response times. The overall response time for a set of DSS queries run serially at a single work station may improve more than 25 percent when SMP support is enabled. The amount of improvement will depend in part on the number of processors participating in each query execution and the optimization algorithms used to implement the query. Some individual queries can see significantly larger gains.

An online course, *DB2 Symmetric Multiprocessing for System i: Database Parallelism within i5/OS,* including a pdf form of the course materials is available at http://www-03.ibm.com/servers/enable/site/education/ibp/4aea/index.html.

## 4.7 DB2 for i5/OS Memory Sharing Considerations

DB2 for i5/OS has internal algorithms to automatically manage and share memory among jobs. This eliminates the complexity of setting and tuning many parameters which are essential to getting good performance on other database products. The memory sharing algorithms within SQE and i5/OS will limit the amount of memory available to execute an SQL query to a 'job share'. The optimizer will choose an access plan which is optimal for the job's share of the memory pool and the query engine will

limit the amount of data it brings into and keeps in memory to a job's share of memory. The amount of memory available to each job is inversely proportional to the number of active jobs in a memory pool.

The memory-sharing algorithms discussed above provide balanced performance for all the jobs running in a memory pool. Running short transactional queries in the same memory pool as long running, data intensive queries is acceptable. However, if it is desirable to get maximum performance for long-running, data-intensive queries it may be beneficial to run these types of queries in a memory pool dedicated to this type of workload. Executing long-running, data-intensive queries in the same memory pool with a large volume of short transactional queries will limit the amount of memory available for execution of the long-running query. The plan choice and engine execution of the long-running query will be tuned to run in the amount of memory comparable to that available to the jobs running the short transactional queries. In many cases, data-intensive, long-running queries will get improved performance with larger amounts of memory. With more memory available the optimizer is able to consider access plans which may use more memory, but will minimize runtime. The query engine will also be able to take advantage of additional memory by keeping more data in memory potentially eliminating a large number of DASD I/Os. Also, for a job executing long-running performance critical queries in a separate pool, it may be beneficial to set QQRYDEGREE=*MAX. This will allow all memory in the pool to be used by the job to process a query. Thus running the longer-running, data intensive queries in a separate pool may dramatically reduce query runtime.

## 4.8 Journaling and Commitment Control

### Journaling

The primary purpose of journal management is to provide a method to recover database files. Additional uses related to performance include the use of journaling to decrease the time required to back up database files and the use of access path journaling for a potentially large reduction in the length of abnormal IPLs. For more information on the uses and management of journals, refer to the *Sytem i Backup and Recovery Guide*. For more detailed information on the performance impact of journaling see the redbook *Striving for Optimal Journal Performance on DB2 Universal Database for System i*.

The addition of journaling to an application will impact performance in terms of both CPU and I/O as the application changes to the journaled file(s) are entered into the journal. Also, the job that is making the changes to the file must wait for the journal I/O to be written to disk, so response time will in many cases be affected as well.

Journaling impacts the performance of each job differently, depending largely on the amount of database writes being done. Applications doing a large number of writes to a journaled file will most likely show a significant degradation both in CPU and response time while an application doing only a limited number of writes to the file may show only a small impact.

### Remote Journal Function

The remote journal function allows replication of journal entries from a local (source) System i to a remote (target) System i by establishing journals and journal receivers on the target system that are associated with specific journals and journal receivers on the source system. Some of the benefits of using remote journal include:

- Allows customers to replace current programming methods of capturing and transmitting journal entries between systems with more efficient system programming methods. This can result in lower CPU consumption and increased throughput on the source system.

- Can significantly reduce the amount of time and effort required by customers to reconcile their source and target databases after a system failure. If the synchronous delivery mode of remote journal is used (where journal entries are guaranteed to be deposited on the target system prior to control being returned to the user application), then there will be no journal entries lost. If asynchronous delivery mode is used, there may be some journal entries lost, but the number of entries lost will most likely be fewer than if customer programming methods were used due to the reduced system overhead of remote journal.

- Journal receiver save operations can be offloaded from the source system to the target system, thus further reducing resource and consumption on the source system.

Hot backup, data replication and high availability applications are good examples of applications which can benefit from using remote journal. Customers who use related or similar software solutions from other vendors should contact those vendors for more information.

**System-Managed Access Path Protection (SMAPP)**

System-Managed Access Path Protection (SMAPP) offers system monitoring of potential access path rebuild time and automatically starts and stops journaling of system selected access paths. In the unlikely event of an abnormal IPL, this allows for faster access path recovery time.

SMAPP does implicit access path journaling which provides for limited partial/localized recovery of the journaled access paths. This provides for much faster IPL recovery steps. An estimation of how long access path recovery will take is provided by SMAPP, and SMAPP provides a setting for the acceptable length of recovery. SMAPP is shipped enabled with a default recovery time. For most customers, the default value will minimize the performance impact, while at the same time provide a reasonable and predictable recovery time and protection for critical access paths. But the overhead of SMAPP will vary from system to system and application to application. As the target access path recovery time is lowered, the performance impact from SMAPP will increase as the SMAPP background tasks have to work harder to meet this target. There is a balance of recovery time requirements vs. the system resources required by SMAPP.

Although SMAPP may start journaling access paths, it is recommended that the most important/large/critical/performance sensitive access paths be journaled explicitly with STRJRNAP. This eliminates the extra overhead of SMAPP evaluating these access paths and implicitly starting journaling for the same access path day after day. A list of the currently protected access paths may be seen as an option from the DSPRCYAP screen. Indexes which consistently show up at the top of this list may be good candidates for explicit journaling via the STRJRNAP command. As identifying important access paths can be a difficult task, SMAPP provides a good safety net to protect those not explicitly journaled.

In addition to the setting to specify a target recovery time, SMAPP also has the following special settings which may be selected with the EDTRCYAP and CHGRCYAP commands:
- *MIN - all exposed indexes will be protected
- *NONE - no indexes will be protected; SMAPP statistics will be maintained
- *OFF - no indexes will be protected; No SMAPP statistics will be maintained (Restricted Mode)

It is highly recommended that SMAPP protection NOT be turned off.

There are 3 sets of tasks which do the SMAPP work. These tasks work in the background at low priority to minimize the impact of SMAPP on system performance. The tasks are as follows:

- JO_EVALUATE-TASK - Evaluates indexes, estimates rebuild time for an index, and may start or stop implicit journaling of an index.
- JO-TUNING-TASK - Periodically wakes up to consider where the user recovery threshold is set and manages which indexes should be implicitly journaled.
- JORECRA-DEF-XXX and JORECRA-USR-XXX tasks are the worker tasks which sweep aged journal pages from main memory to minimize the amount of recovery needed during IPL.

Here are guidelines for lowering the amount of work for each of these tasks:

- If the JO-TUNING-TASK seems busy, you may want to increase SMAPP recovery target time.
- If the JO-EVALUATE task seems busy, explicitly journaling the largest access paths may help or looks for jobs that are opening/closing files repeatedly.
- If the JORECRA tasks seem busy, you may want to increase journal recovery ratio.
- Also, if the target recovery time is not being met there may be SMAPP ineligible access paths. These should be modified so as to become SMAPP eligible.

To monitor the performance impacts of SMAPP there are Performance Explorer trace points and a substantial set of Collection Services counters which provide information on the SMAPP work.

SMAPP makes a decision of where to place the implicit access path journal entries. If the underlying physical file is not journaled, SMAPP will place the entries in a default (hidden) system journal. If the underlying physical file is journaled, SMAPP will place the implicit journal entries in the same place. SMAPP automatically manages the system journal. For the user journal receivers used by SMAPP, RCVSIZOPT(*RMVINTENT), as specified on the CHGJRN command, is a recommended option. The disk space used by SMAPP may be displayed with the EDTRCYAP and DSPRCYAP commands. It rarely exceeds 1% of the ASP size.

For more information on SMAPP see the Systems management -> Journal management -> System-managed access path protection section in the System i information center.

### Commitment Control

Commitment control is an extension to the journal function that allows users to ensure that all changes to a transaction are either all complete or, if not complete, can be easily backed out. The use of commitment control adds two more journal entries, one at the beginning of the committed transaction and one at the end, resulting in additional CPU and I/O overhead. In addition, the time that record level locks are held increases with the use of commitment control. Because of this additional overhead and possible additional record lock contention, adding commitment control will in many cases result in a noticeable degradation in performance for an application that is currently doing journaling.

---

## 4.9 DB2 Multisystem for i5/OS

DB2 Multisystem for i5/OS offers customers the ability to distribute large databases across multiple System i servers in order to gain nearly unlimited scalability and improved performance for many large query operations. Multiple System i servers are coupled together in a shared-nothing cluster where each system uses its own main memory and disk storage. Once a database is properly partitioned among the

multiple nodes in the cluster, access to the database files is seamless and transparent to the applications and users that reference the database. To the users, the partitioned files still behave as though they were local to their system.

The most important aspect of obtaining optimal performance with DB2 Multisystem is to plan ahead for what data should be partitioned and how it should be partitioned. The main idea behind this planning is to ensure that the systems in the cluster run in parallel with each other as much as possible when processing distributed queries while keeping the amount of communications data traffic to a minimum. Following is a list of items to consider when planning for the use of distributed data via DB2 Multisystem.

- Avoid large amounts of data movement between systems. A distributed query often achieves optimal performance when it is able to divide the query among several nodes, with each node running its portion of the query on data that is local to that system and with a minimum number of accesses to remote data on other systems. Also, if a file that is heavily used for transaction processing is to be distributed, it should be done such that most of the database accesses are local since remote accesses may add significantly to response times.

- Choosing which files to partition is important. The largest improvements will be for queries on large files. Files that are primarily used for transaction processing and not much query processing are generally not good candidates for partitioning. Also, partitioning files with only a small number of records will generally not result in much improvement and may actually degrade performance due to the added communications overhead.

- Choose a partitioning key that has many different values. This will help ensure a more even distribution of the data across the multiple nodes. In addition, performance will be best if the partitioning key is a single field that is a simple data type.

- It is best to choose a partition key that consists of a field or fields whose values are not updated. Updates on partition keys are only allowed if the change to the field(s) in the key will not cause that record to be partitioned to a different node.

- If joins are often performed on multiple files using a single field, use that field as the partitioning key for those files. Also, the fields used for join processing should be of the same data type.

- It will be helpful to partition the database files based on how quickly each node can process its portion of the data when running distributed queries. For example, it may be better to place a larger amount of data on a large multiprocessor system than on a smaller single processor system. In addition, current normal utilization levels of other resources such as main memory, DASD and IOPs should be considered on each system in order to ensure that no one individual system becomes a bottleneck for distributed query performance.

- For the best query performance involving distributed files, avoid the use of commitment control when possible. DB2 Multisystem uses two-phase commit, which can add a significant amount of overhead when running distributed queries.

For more information on DB2 Multisystem refer to the DB2 Multisystem manual.

## 4.10 Referential Integrity

In a database user environment, there are frequent cases where the data in one file is dependent upon the data in another file. Without support from the database management system, each application program that updates, deletes or adds new records to the files must contain code that enforces the data dependency rules between the files. Referential Integrity (RI) is the mechanism supported by DB2 that offers its users the ability to enforce these rules without specifically coding them in their application(s). The data dependency rules are implemented as referential constraints via either CL commands or SQL statements that are available for adding, removing and changing these constraints.

For those customers that have implemented application checking to maintain integrity of data among files, there may be a noticeable performance gain when they change the application to use the referential integrity support. The amount of improvement depends on the extent of checking in the existing application. Also, the performance gain when using RI may be greater if the application currently uses SQL statements instead of HLL native database support to enforce data dependency rules.

When implementing RI constraints, customers need to consider which data dependencies are the most commonly enforced in their applications. The customer may then want to consider changing one or more of these dependencies to determine the level of performance improvement prior to a full scale implementation of all data dependencies via RI constraints.

For more information on Referential Integrity see the chapter *Ensuring Data Integrity with Referential Constraints* in *DB2 Universal Database for System i Database Programming* manual and the redbook *Advanced Functions and Administration on DB2 Universal Database for System i*.

## 4.11 Triggers

Trigger support for DB2 allows a user to define triggers (user written programs) to be called when records in a file are changed. Triggers can be used to enforce consistent implementation of business rules for database files without having to add the rule checking in all applications that are accessing the files. By doing this, when the business rules change, the user only has to change the trigger program.

There are three different types of events in the context of trigger programs: insert, update and delete. Separate triggers can be defined for each type of event. Triggers can also be defined to be called before or after the event occurs.

Generally, the impact to performance from applying triggers on the same system for files opened without commitment control is relatively low. However, when the file(s) are under commitment control, applying triggers can result in a significant impact to performance.

Triggers are particularly useful in a client server environment. By defining triggers on selected files on the server, the client application can cause synchronized, systematic update actions to related files on the server with a single request. Doing this can significantly reduce communications traffic and thus provide noticeably better performance both in terms of response time and CPU. This is true whether or not the file is under commitment control.

The following are performance tips to consider when using triggers support:

- Triggers are activated by an external call. The user needs to weigh the benefit of the trigger against the cost of the external call.

- If a trigger is going to be used, leave as much validation to the trigger program as possible.

- Avoid opening files in a trigger program under commitment control if the trigger program does not cause changes to commitable resources.

- Since trigger programs are called repeatedly, minimize the cost of program initialization and unneeded repeated actions. For example, the trigger program should not have to open and close a file every time it is called. If possible, design the trigger program so that the files are opened during the first call and stay open throughout. To accomplish this, avoid SETON LR in RPG, STOP RUN in COBOL and exit() in C.

- If the trigger program opens a file multiple times (perhaps in a program which it calls), make use of shared opens whenever possible.

- If the trigger program is written for the Integrated Language Environment (ILE), make sure it uses the caller's activation group. Having to start a new activation group every time the time the trigger program is called is very costly.

- If the trigger program uses SQL statements, it should be optimized such that SQL makes use of reusable ODPs.

In conclusion, the use of triggers can help enforce business rules for user applications and can possibly help improve overall system performance, particularly in the case of applying changes to remote systems. However, some care needs to be used in designing triggers for good performance, particularly in the cases where commitment control is involved. For more information see the redbook *Stored Procedures, Triggers and User Defined Functions on DB2 Universal Database for System i*.

---

## 4.12 Variable Length Fields

Variable length field support allows a user to define any number of fields in a file as variable length, thus potentially reducing the number of bytes that need to be stored for a particular field.

### Description

Variable length field support on i5/OS has been implemented with a spill area, thus creating two possible situations: the non-spill case and the spill case. With this implementation, when the data overflows, all of the data is stored in the spill portion. An example would be a variable length field that is defined as having a maximum length of 50 bytes and an allocated length of 20 bytes. In other words, it is expected that the majority of entries in this field will be 20 bytes or less and occasionally there will be a longer entry up to 50 bytes in length. When inserting an entry that has a length of 20 bytes or less that entry will be inserted into the allocated part of the field. This is an example of a non-spill case. However, if an entry is inserted that is, for example, 35 bytes long, all 35 bytes will go into the spill area.

To create the variable length field just described, use the following DB2 statement:

```
CREATE TABLE library/table-name
       (field VARCHAR(50) ALLOCATE(20) NOT NULL)
```

In this particular example the field was created with the NOT NULL option. The other two options are NULL and NOT NULL WITH DEFAULT. Refer to the NULLS section in the SQL Reference to determine which NULLS option would be best for your use. Also, for additional information on variable length field support, refer to either the SQL Reference or the SQL Programming Concepts.

**Performance Expectations**

- Variable length field support, when used correctly, can provide performance improvements in many environments. The savings in I/O when processing a variable length field can be significant. The biggest performance gains that will be obtained from using variable length fields are for description or comment types of fields that are converted to variable length. However, because there is additional overhead associated with accessing the spill area, it is generally not a good idea to convert a field to variable length if the majority (70-100%) of the records would have data in this area. To avoid this problem, design the variable length field(s) with the proper allocation length so that the amount of data in the spill area stays below the 60% range. This will also prevent a potential waste of space with the variable length implementation.

- Another potential savings from the use of variable length fields is in DASD space. This is particularly true in implementations where there is a large difference between the ALLOCATE and the VARCHAR attributes AND the amount of spill data is below 60%. Also, by minimizing the size of the file, the performance of operations such as CPYF (Copy File) will also be improved.

- When using a variable length field as a join field, the impact to performance for the join will depend on the number of records returned and the amount of data that spills. For a join field that contains a low percentage of spill data and which already has an index built over it that can be used in the join, a user would most likely find the performance acceptable. However, if an index must be built and/or the field contains a large amount of overflow, a performance problem will likely occur when the join is processed.

- Because of the extra processing that is required for variable length fields, it is not a good idea to convert every field in a file to variable length. This is particularly true for fields that are part of an index key. Accessing records via a variable length key field is noticeably slower than via a fixed length key field. Also, index builds over variable length fields will be noticeably slower than over fixed length fields.

- When accessing a file that contains variable length fields through a high-level language such as COBOL, the variable that the field is read into must be defined as variable or of a varying length. If this is not done, the data that is read in to the fixed length variable will be treated as fixed length. If the variable is defined as PIC X(40) and only 25 bytes of data is read in, the remaining 15 bytes will be space filled. The value in that variable will now contain 40 bytes. The following COBOL example shows how to declare the receiving variable as a variable length variable:

```
01 DESCR.
       49 DESCR-LEN      PIC S9(4) COMP-4.
       49 DESCRIPTION    PIC X(40).

    EXEC SQL
        FETCH C1 INTO DESCR
    END-EXEC.
```

For more detail about the vary-length character string, refer to the SQL Programmer's Guide.

The above point is also true when using a high-level language to insert values into a variable length field. The variable that contains the value to be inserted must be declared as variable or varying. A PL/I example follows:

```
DCL FLD1 CHAR(40) VARYING;
FLD1 = XYZ Company;

EXEC SQL
    INSERT INTO library/file VALUES
        ("001453", FLD1, ...);
```

Having defined FLD1 as VARYING will, for this example, insert a data string of 11 bytes into the field corresponding with FLD1 in this file. If variable FLD1 had not been defined as VARYING, a data string of 40 bytes would be inserted into the corresponding field. For additional information on the VARYING attribute, refer to the PL/I User's Guide and Reference.

- In summary, the proper implementation and use of DB2 variable length field support can help provide overall improvements in both function and performance for certain types of database files. However, the amount of improvement can be greatly impacted if the new support is not used correctly, so users need to take care when implementing this function.

---

## 4.13 Reuse Deleted Record Space

### Description of Function

This section discusses the support for reuse of deleted record space. This database support provides the customer a way of placing newly-added records into previously deleted record spaces in physical files. This function should reduce the requirement for periodic physical file reorganizations to reclaim deleted record space. File reorganization can be a very time consuming process depending on the size of the file and the number of indexes over it, along with the reorganize options selected. To activate the reuse function, set the Reuse deleted records (REUSEDLT) parameter to *YES on the CRTPF (Create Physical File) The default value when creating a file with CRTPF is *NO (do not reuse). The default for SQL Create Table is *YES.

### Comparison to Normal Inserts

Inserts into deleted record spaces are handled differently than normal inserts and have different performance characteristics. For normal inserts into a physical file, the database support will find the end of the file and seize it once for exclusive use for the subsequent adds. Added records will be written in blocks at the end of the file. The size of the blocks written will be determined by the default block size or by the size specified using an Override Database File (OVRDBF) command. The SEQ(*YES number of records) parameter can be used to set the block size.

In contrast, when reuse is active, the database support will process the added record more like an update operation than an add operation. The database support will maintain a bit map to keep track of deleted records and to provide fast access to them. Before a record can be added, the database support must use the bit-map to find the next available deleted record space, read the page containing the deleted record entry into storage, and seize the deleted record to allow replacement with the added record. Lastly, the added records are blocked as much as permissible and then written to the file.

To summarize, additional CPU processing will be required when reuse is active to find the deleted records, perform record level seizes and maintain the bit-map of deleted records. Also, there may be some additional disk I/O required to read in the deleted records prior to updating them. However, this extra overhead is generally less than the overhead associated with a sequential update operation.

**Performance Expectations**

The impact to performance from implementing the reuse deleted records function will vary depending on the type of operation being done. Following is a summary of how this function will affect performance for various scenarios:

- When blocking was not specified, reuse was slightly faster or equivalent to the normal insert application. This is due to the fact that reuse by default blocks up records for disk I/Os as much as possible.

- Increasing the number of indexes over a file will cause degradation for all insert operations, regardless of whether reuse is used or not. However, with reuse activated, the degradation to insert operations from each additional index is generally higher than for normal inserts.

- The RGZPFM (Reorganize Physical File Member) command can run for a long period of time, depending on the number of records in the file and the number of indexes over the file and the chosen command options. Even though activating the reuse function may cause some performance degradation, it may be justified when considering reorganization costs to reclaim deleted record space.

- The reuse function can always be deactivated if the customer encounters a critical time window where no degradation is permissible. The cost of activating/de-activating reuse is relatively low in most cases.

- Because the reuse function can lead to smaller sized files, the performance of some applications may actually improve, especially in cases where sequential non-keyed processing of a large portion of the file(s) is taking place.

## 4.14 Performance References for DB2

1. The home page for DB2 Universal Database for System i is found at
   http://www-1.ibm.com/servers/eserver/iseries/db2/
   This web site includes the recent announcement information, white paper and technical articles, and DB2 education information.

2. The System i information center section on *DB2 for i5/OS* under *Database and file systems* has information on all aspects of DB2 for i5/OS including the section *Monitor and Tune database* under *Administrative topics*. This can be found at url: http://www.ibm.com/eserver/iseries/infocenter

3. Information on creating efficient running queries and query performance monitoring and tuning is found in the DB2 for i5/OS *Database Performance and Query Optimization* manual. This document contains detailed information on access methods, the query optimizer, and optimizing query performance including using database monitor to monitor queries, using QAQQINI file options and using indexes. To access this document look in the Printable PDF section in the System i information center.

4. The System i redbooks provide performance information on a variety of topics for DB2. The redbook repository is located at http://publib-b.boulder.ibm.com/Redbooks.nsf/portals/systemi.

# Chapter 5. Communications Performance

There are many factors that affect System i performance in a communications environment. This chapter discusses some of the common factors and offers guidance on how to help achieve the best possible performance. Much of the information in this chapter was obtained as a result of analysis experience within the Rochester development laboratory. Many of the performance claims are based on supporting performance measurement and analysis with the NetPerf and Netop workloads. In some cases, the actual performance data is included here to reinforce the performance claims and to demonstrate capacity characteristics. The NetPerf and Netop workloads are described in section 5.2.

This chapter focuses on communication in non-secure and secure environments on Ethernet solutions using TCP/IP. Many applications require network communications to be secure. Communications and cryptography, in these cases, must be considered together. Secure Socket Layer (SSL), Transport Layer Security (TLS) and Virtual Private Networking (VPN) capacity characteristics will be discussed in section 5.5 of this chapter. For information about how the Cryptographic Coprocessor improves performance on SSL/TLS connections, see section 8.4 of Chapter 8, "Cryptography Performance."

**Communications Performance Highlights for IBM i Operation System 5.4:**

- The support for the new Internet Protocol version 6 (IPv6) has been enhanced. The new IPv6 functions are consistent at the product level with their respective IPv4 counterparts.

- Support is added for the 10 Gigabit Ethernet optical fiber input/output adapters (IOAs) 573A and 576A. These IOAs do not require an input/output processor (IOP) to be installed in conjunction with the IOA. Instead the IOA can be plugged into a PCI bus slot and the IOA is controlled by the main processor. The 573A is a 10 Gigabit SR (short reach) adapter, which uses multimode fiber (MMF) and has a duplex LC connector. The 573A can transmit to lengths of 300 meters. The 576A is a 10 Gigabit LR (long reach) adapter, which uses single mode fiber (SMF) and has a duplex SC connector. The 576A can transmit to lengths of 10 kilometers. Both of these adapters support TCP/IP, 9000-byte jumbo frames, checksum offloading and the IEEE 802.3ae standard.

- The IBM 5706 2-Port 10/100/1000 Base-TX PCI-X IOA and IBM 5707 2-Port Gigabit Ethernet-SX PCI-X IOA supports checksum offloading and 9000-byte jumbo frames (1 Gigabit only). These adapters do not require an IOP to be installed in conjunction with the IOA.

- The IBM 5701 10/100/1000 Base-TX PCI-X IOA does not require an IOP to be installed in conjunction with the IOA.

- The IBM Cryptographic Access Provider product, 5722-AC3 (128-bit) is no longer required. This is a new development for the 5.4 release of IBM i Operation System. All 5.4 systems are capable of the function that was previously provided in the 5722-AC3 product. This is relevant for SSL communications.

**Communications Performance Highlights for IBM i Operation System 5.4.5:**

- The IBM 5767 2-Port 10/100/1000 Based-TX PCI-E IOA and IBM 5768 2-Port Gigabit Ethernet-SX PCI-E IOA supports checksum offloading and 9000-byte jumbo frames (1 Gigabit only). These adapters do not require an IOP to be installed in conjunction with the IOA.

- IBM's Host Ethernet Adapter (HEA) integrated 2-Port 10/100/1000 Based-TX PCI-E IOA supports checksum offloading, 9000-byte jumbo frames (1 Gigabit only) and LSO - Large Send Offload (IPv4 only). These adapters do not require an IOP to be installed in conjunction with the IOA. Additionally, each physical port has 16 logical ports that may be assigned to other partitions and allows each partition to utilize the same physical port simultaneously with the following limitation: one logical port, per physical port, per partition.

**Communications Performance Highlights for IBM i Operation System 6.1:**

- Additional enhancement in Internet Protocol version 6 (IPv6) in the following areas:
  1. Advanced Sockets APIs
  2. Path MTU Discovery
  3. Correspondent Node Mobility Support
  4. Support of Privacy extensions to stateless address auto-configuration
  5. Virtual IP address,
  6. Multicast Listener Discovery v2 support
  7. Router preferences and more specific route advertisement support
  8. Router load sharing.

- Additional enhancement in Internet Protocol version 4 (IPv4) in the following areas:
  1. Remote access proxy fault tolerance
  2. IGMP v3 support for IPv4 multicast.

- Large Send Offload support was implemented for Host Ethernet Adapter ports on Internet Protocol version 4 (IPv4).

## 5.1 System i Ethernet Solutions

The need for communication between computer systems has grown over the last decades, and TCP/IP over Ethernet has grown with it. We currently have arrived where different factors influence the capabilities of the Ethernet. Some of these influences can come from the cabling and adapter type chosen. Limiting factors can be the capabilities of the hub or switch used, the frame size you are able to transmit and receive, and the type of connection used. The System i server is capable of transmitting and receiving data at speeds of 10 megabits per second (10 Mbps) to 10 gigabits per second (10 Gbps or 10 000 Mbps) using an Ethernet IOA. Functions such as full duplex also enhance the communication speeds and the overall performance of Ethernet.

Table 5.1 contains a list of Ethernet input/output adapters that are used to create the results in this chapter.

*Table 5.1*

**Ethernet input/output adapters**

| CCIN[3] | Description | Speed[6] (Mbps) | Jumbo frames supported | Operations Console supported | Duplex mode capability | |
|---|---|---|---|---|---|---|
| | | | | | Full | Half |
| 2849[1] | 10/100 Mbps Ethernet | 10 / 100 | No | Yes | Yes | Yes |
| 5700[2] | IBM Gigabit Ethernet-SX PCI-X | 1000 | Yes | No | Yes | No |
| 5701[1] | IBM 10/100/1000 Base-TX PCI-X | 10 / 100 / 1000 | Yes | No | Yes | Yes |
| 5706[1] | IBM 2-Port 10/100/1000 Base-TX PCI-X[7] | 10 / 100 / 1000 | Yes | Yes | Yes | Yes |
| 5707[2] | IBM 2-Port Gigabit Ethernet-SX PCI-X[7] | 1000 | Yes | Yes | Yes | No |
| 5767[1] | IBM 2-Port 10/100/1000 Base-TX PCI-e[7] | 10 / 100 / 1000 | Yes | Yes | Yes | Yes |
| 5768[2] | IBM 2-Port Gigabit Ethernet-SX PCI-e[7] | 1000 | Yes | Yes | Yes | No |
| 573A[2] | IBM 10 Gigabit Ethernet-SX PCI-X | 10000 | Yes | No | Yes | No |

| | | | | | | |
|---|---|---|---|---|---|---|
| 181A[1] | IBM 2-Port 10/100/1000 Base-TX PCI-e[7] | 10 / 100 / 1000 | Yes | Yes | Yes | Yes |
| 181B[2] | IBM 2-Port Gigabit Base-SX PCI-e | 10000 | Yes | Yes | Yes | Yes |
| 181C[1] | IBM 4-Port 10/100/1000 Base-TX PCI-e[7] | 10 / 100 / 1000 | Yes | Yes | Yes | Yes |
| 1819[1] | IBM 4-Port 10/100/1000 Base-TX PCI-e[7,9] | 10 / 100 / 1000 | Yes | Yes | Yes | Yes |
| N/A | Virtual Ethernet[4] | n/a[5] | Yes | N/A | Yes | No |
| N/A | Blade[8] | n/a[5] | Yes | N/A | Yes | Yes |

**Notes:**

1. Unshielded Twisted Pair (UTP) card; uses copper wire cabling
2. Uses fiber optics
3. Custom Card Identification Number and System i Feature Code
4. Virtual Ethernet enables you to establish communication via TCP/IP between logical partitions and can be used without any additional hardware or software.
5. Depends on the hardware of the system.
6. These are theoretical hardware unidirectional speeds
7. Each port can handle 1000 Mbps
8. Blade communicates with the VIOS Partition via Virtual Ethernet
9. Host Ethernet Adapter for IBM Power 550, 9409-M50 running IBM i Operating System

- All adapters support Auto-negotiation

## 5.2 Communication Performance Test Environment

### Hardware

All PCI-X measurements for 100 Mbps and 1 Gigabit were completed on an IBM System i 570+ 8-Way (2.2 GHz). Each system is configured as an LPAR, and each communication test was performed between two partitions on the same system with one dedicated CPU. The gigabit IOAs were installed in a 133MHz PCI-X slot.

The measurements for 10 Gigabit were completed on two IBM System i 520+ 2-Way (1.9 GHz) servers. Each System i server is configured as a single LPAR system with one dedicated CPU. Each communication test was performed between the two systems and the 10 Gigabit IOAs were installed in the 266 MHz PCI-X DDR(double data rate) slot for maximum performance. Only the 10 Gigabit Short Reach (573A) IOA's were used in our test environment.

All PCI-e measurements were completed on an IBM System i 9406-MMA 7061 16 way or IBM Power 550, 9409-M50. Each system is configured as an LPAR, and each communication test was performed between two partitions on the same system with one dedicated CPU. The Gigabit IOA's where installed in a PCI-e 8x slot.

All Blade Center measurements where collected on a 4 processor 7998-61X Blade in a Blade Center H chassis, 32 GB of memory. The AIX partition running the VIOS server was not limited. All performance data was collect with the Blade running as the server. The System i partition (on the Blade) was limited to 1 CPU with 4 GB of memory and communicated with an external IBM System i 570+ 8-Way (2.2 GHz) configured as a single LPAR system with one dedicated CPU and 4 GB of Memory.

### Software

The NetPerf and Netop workloads are primitive-level function workloads used to explore communications performance. Workloads consist of programs that run between a System i client and a System i server, Multiple instances of the workloads can be executed over multiple connections to increase the system load. The programs communicate with each other using sockets or SSL APIs.

To demonstrate communications performance in various ways, several workload scenarios are analyzed. Each of these scenarios may be executed with regular nonsecure sockets or with secure SSL using the GSK API:

1. **Request/Response (RR):** The client and server send a specified amount of data back and forth over a connection that remains active.
2. **Asymmetric Connect/Request/Response (ACRR)**: The client establishes a connection with the server, a single small request (64 bytes) is sent to the server, and a response (8K bytes) is sent by the server back to the client, and the connection is closed.
3. **Large transfer (Stream):** The client repetitively sends a given amount of data to the server over a connection that remains active.

The NetPerf and Netop tools used to measure these benchmarks merely copy and transfer the data from memory. Therefore, additional consideration must be given to account for other normal application processing costs (for example, higher CPU utilization and higher response times due to disk access time). A real user application will have this type of processing as only a percentage of the overall workload. The IBM Systems Workload Estimator, described in Chapter 23, reflects the performance of real user applications while averaging the impact of the differences between the various communications protocols. The real world perspective offered by the Workload Estimator can be valuable for projecting overall system capacity.

## 5.3 Communication and Storage observations

With the continued progress in both communication and storage technology, it is possible that the performance bottleneck shifts. Especially with high bandwidth communication such as 10 Gigabit and Virtual ethernet, storage technology could become the limiting factor.

### DASD Performance

Storage performance is dependent on the configuration and amount of disk units within your partition. Table 14.1.2.2 in chapter 14. DASD Performance shows this for save and restore operations for 2 different IOA's. See the chapter for detailed information.

*Table 5.2 - Copy of Table 14.1.2.2 in chapter 14. DASD Performance*

| IOA and operation | | Number of 35 GB DASD units (Measurement numbers in GB/HR) | | |
|---|---|---|---|---|
| 2778 IOA | | 15 Units | 30 Units | 45 Units |
| *SAVF | Save | 41 | 83 | 122 |
| | Restore | 41 | 83 | 122 |
| 2757 IOA | | | | |
| *SAVF | Save | 82 | 165 | 250 |
| | Restore | 82 | 165 | 250 |

### Large data transfer (FTP)

When transferring large amounts of data, for example with FTP, DASD performance plays an important role. Both the sending and receiving end could limit the communication speed when using high bandwidth communication. Also in a multi-threading environment, having more then one streaming session could improve overall communication performance when the DASD throughput is available.

*Table 5.3*

| Virtual Ethernet | Performance in MB per second | |
|---|---|---|
| **FTP** | **1 Disk Unit ASP on 2757 IOA** | **15 Disk Units ASP on 2757 IOA** |
| 1 Session | 10.8 | 42.0 |
| 2 Sessions | 10.5 | 70.0 |
| 3 Sessions | 10.4 | 75.0 |
| | | |

## 5.4 TCP/IP non-secure performance

In table 5.4 you will find the payload information for the different Ethernet types. The most important factor with streaming is to determine how much data can be transferred. The results are listed in bits and bytes per second. Virtual Ethernet does not have a raw bit rate, since the maximum throughput is determined by the CPU.

*Table 5.4*

**Streaming Performance**

| Ethernet Type | Raw bit rate[1] (Mbits per second) | MTU[2] | Payload Simplex[3] (Mbits per second) | Payload Duplex[4] (Mbits per second) |
|---|---|---|---|---|
| 100 Megabit | 100 | 1,492 | 93.5 | 170.0 |
| 1 Gigabit | 1,000 | 1,492 | 935.4 | 1740.3 |
| | | 8,992 | 935.9 | 1753.1 |
| 10 Gigabit[5] | 10,000 | 1,492 | 3745.4 | 4400.7 |
| | | 8,992 | 8789.6 | 9297.0 |
| HEA 1 Gigabit | 1,000 | 1,492 | 986.4 | 1481.4 |
| | | 8,992 | 941.1 | 1960.9 |
| | 160,00[7] | 1,492 | 2811.8 | 6331.0 |
| | | 8,992 | 9800.7 | 10586.4 |
| HEA 10 Gigabit | 10,000 | 1,492 | 2913.1 | 3305.2 |
| | | 8,992 | 9392.3 | 9276.9 |
| | 160,00[7] | 1,492 | 2823.5 | 6332.3 |
| | | 8,992 | 9813.7 | 10602.3 |
| Blade[8] | n/a | 1,492 | 933.1 | 1014.4 |
| Virtual[6] | n/a | 8,992 | 8553.0 | 11972.3 |

**Notes:**
1. The Raw bit rate value is the physical media bit rate and does not reflect physical media overheads
2. Maximum Transmission Unit. The large (8992 bytes) MTU is also referred to as Jumbo Frames.
3. Simplex is a single direction TCP data stream.
4. Duplex is a bidirectional TCP data stream.
5. The 10 Gigabit results were obtained by using multiple sessions, because a single sessions is incapable to fully utilize the 10 Gigabit adapter.
6. Virtual Ethernet uses Jumbo Frames only, since large packets are supported throughout the whole connection path.
7. HEA P.P.U.T (Partition to Partition Unicast Traffic or internal switch) 16 Gbps per port group.
8. 4 Processor 7998-61X Blade
9. All measurements are performed with Full Duplex Ethernet.

Streaming data is not the only type of communication handled through Ethernet. Often server and client applications communicate with small packets of data back and forth (RR). In the case of web browsers, the most common type is to connect, request and receive data, then disconnect (ACRR). Table 5.5 provides some rough capacity planning information for these RR and ACRR communications.

*Table 5.5*

| RR & ACRR Performance<br>(Transactions per second per server CPU) | | | |
|---|---|---|---|
| **Transaction Type** | **Threads** | **1 Gigabit** | **Virtual** |
| Request/Response (RR) 128 Bytes | 1 | 991.32 | 873.62 |
| | 26 | 1330.45 | 912.34 |
| Asym. Connect/Request/Response (ACRR) 8K Bytes | 1 | 261.51 | 218.82 |
| | 26 | 279.64 | 221.21 |

**Notes**:

- Capacity metrics are provided for nonsecure transactions
- The table data reflects System i as a server (not a client)
- The data reflects Sockets and TCP/IP
- This is only a rough indicator for capacity planning. Actual results may differ significantly.
- All measurement where taken with Packet Trainer off (See 5.6 for line dependent performance enhancements)

Here the results show the difference in performance for different Ethernet cards compared with Virtual Ethernet. We also added test results with multiple threads to give an insight on the performance when a system is stressed with multiple sessions.

This information is of similar type to that provided in Chapter 6, Web Server Performance. There are also capacity planning examples in that chapter.

---

## 5.5 TCP/IP Secure Performance

With the growth of communication over public network environments like the Internet, securing the communication data becomes a greater concern. Good examples are customers providing personal data to complete a purchase order (SSL) or someone working away from the office, but still able to connect to the company network (VPN).

**SSL**

SSL was created to provide a method of session security, authentication of a server or client, and message authentication. SSL is most commonly used to secure web communication, but SSL can be used for any reliable communication protocol (such as TCP). The successor to SSL is called TLS. There are slight differences between SSL v3.0 and TLS v1.0, but the protocol remains substantially the same. For the data gathered here we only use the TLS v1.0 protocol. Table 5.6 provides some rough capacity planning information for SSL communications, when using 1 Gigabit Ethernet.

*Table 5.6*

| | SSL Performance (transactions per second per server CPU) | | | | | |
|---|---|---|---|---|---|---|
| **Transaction Type:** | **Nonsecure TCP/IP** | **RC4 / MD5** | **RC4 / SHA-1** | **AES128 / SHA-1** | **AES256 / SHA-1** | **TDES / SHA-1** |
| Request/Response (RR) 128 Byte | 1167 | 565.4 | 530.0 | 479.6 | 462.1 | 202.2 |
| Asym. Connect/Request/Response (ACRR) 8K Bytes | 249.7 | 53.4 | 48.0 | 31.3 | 27.4 | 4.8 |
| Large Transfer (Stream) 16K Bytes | 478.4 | 55.7 | 53.3 | 36.9 | 31.9 | 6.5 |

**Notes**:

- Capacity metrics are provided for nonsecure and each variation of security policy
- The table data reflects System i as a server (not a client)
- This is only a rough indicator for capacity planning. Actual results may differ significantly.
- Each SSL connection was established with a 1024 bit RSA handshake.

This table gives an overview on performance results on using different encryption methods in SSL compared to regular TCP/IP. The encryption methods we used range from fast but less secure (RC4 with MD5) to the slower but more secure (AES or TDES with SHA-1).

With SSL there is always a fixed overhead, such as the session handshake. The variable overhead is based on the number of bytes that need to be encrypted/decrypted, the size of the public key, the type of encryption, and the size of the symmetric key.

These results may be used to estimate a system's potential transaction rate at a given CPU utilization assuming a particular workload and security policy. Say the result of a given test is 5 transactions per second per server CPU. Then multiplying that result with 50 will tell that at 50% CPU utilization a transaction rate of 250 transactions per second is possible for this type of SSL communication on this environment. Similarly when a capacity of 100 transactions per second is required, the CPU utilization can be approximated by dividing 100 by 5, which gives a 20% CPU utilization in this environment. These are only estimations on how to size the workload, since actual results might vary. Similar information about SSL capacity planning can be found in Chapter 6, Web Server Performance.

Table 5.7 below illustrates relative CPU consumption for SSL instead of potential capacity. Essentially, this is a normalized inverse of the CPU capacity data from Table 5.6. It gives another view of the impact of choosing one security policy over another for various NetPerf scenarios.

*Table 5.7*

| | SSL Relative Performance (scaled to Nonsecure baseline) | | | | | |
|---|---|---|---|---|---|---|
| **Transaction Type:** | **Nonsecure TCP/IP** | **RC4 / MD5** | **RC4 / SHA-1** | **AES128 / SHA-1** | **AES256 / SHA-1** | **TDES / SHA-1** |
| Request/Response (RR) 128 Byte | 1.0 x | 2.1 | 2.2 | 2.4 | 2.5 | 5.8 |
| Asym. Connect/Request/Response (ACRR) 8K Bytes | 1.0 y | 4.7 | 5.2 | 8.0 | 9.1 | 51.7 |
| Large Transfer (Stream) 16K Bytes | 1.0 z | 8.6 | 9.0 | 13.0 | 15.0 | 73.7 |

**Notes**:

- Capacity metrics are provided for nonsecure and each variation of security policy
- The table data reflects System i as a server (not a client)
- This is only a rough indicator for capacity planning. Actual results may differ significantly.
- Each SSL connections was established with a 1024 bit RSA handshake.
- x, y and z are scaling constants, one for each NetPerf scenario.

**VPN**

Although the term Virtual Private Networks (VPN) didn't start until early 1997, the concepts behind VPN started around the same time as the birth of the Internet. VPN creates a secure tunnel to communicate from one point to another using an unsecured network as media. Table 5.8 provides some rough capacity planning information for VPN communication, when using 1 Gigabit Ethernet.

*Table 5.8*

| | VPN Performance (transactions per second per server CPU) | | | | |
|---|---|---|---|---|---|
| **Transaction Type:** | **Nonsecure TCP/IP** | **AH with MD5** | **ESP with RC4 / MD5** | **ESP with AES128 / SHA-1** | **ESP with TDES / SHA-1** |
| Request/Response (RR) 128 Byte | 1167.0 | 428.5 | 322.9 | 307.71 | 148.4 |
| Asym. Connect/Request/Response (ACRR) 8K Bytes | 249.7 | 49.9 | 37.7 | 32.7 | 9.1 |
| Large Transfer (Stream) 16K Bytes | 478.4 | 44.0 | 31.0 | 25.6 | 5.4 |

**Notes**:

- Capacity metrics are provided for nonsecure and each variation of security policy
- The table data reflects System i as a server (not a client)
- VPN measurements used transport mode, TDES, AES128 or RC4 with 128-bit key symmetric cipher and MD5 message digest with RSA public/private keys. VPN antireplay was disabled.
- This is only a rough indicator for capacity planning. Actual results may differ significantly.

This table also shows a range of encryption methods to give you an insight on the performance between less secure but faster, or more secure but slower methods, all compared to unsecured TCP/IP.

Table 5.9 below illustrates relative CPU consumption for VPN instead of potential capacity. Essentially, this is a normalized inverse of the CPU capacity data from Table 5.6. It gives another view of the impact of choosing one security policy over another for various NetPerf scenarios.

*Table 5.9*

| | VPN Relative Performance (scaled to Nonsecure baseline) | | | | |
|---|---|---|---|---|---|
| **Transaction Type:** | **Nonsecure TCP/IP** | **AH with MD5** | **ESP with RC4 / MD5** | **ESP with AES128 / SHA-1** | **ESP with TDES / SHA-1** |
| Request/Response (RR) 128 Byte | 1.0 x | 2.7 | 3.6 | 3.8 | 7.9 |
| Asym. Connect/Request/Response (ACRR) 8K Bytes | 1.0 y | 5.0 | 6.6 | 7.6 | 27.5 |
| Large Transfer (Stream) 16K Bytes | 1.0 z | 10.9 | 15.4 | 18.7 | 88.8 |

**Notes**:

- Capacity metrics are provided for nonsecure and each variation of security policy
- The table data reflects System i as a server (not a client)
- VPN measurements used transport mode, TDES, AES128 or RC4 with 128-bit key symmetric cipher and MD5 message digest with RSA public/private keys. VPN anti-replay was disabled.
- This is only a rough indicator for capacity planning. Actual results may differ significantly.
- x, y and z are scaling constants, one for each NetPerf scenario.

The SSL and VPN measurements are based on a specific set of cipher methods and public key sizes. Other choices will perform differently.

---

## 5.6 Performance Observations and Tips

- Communication performance on Blades may see an increase when the processors are in shared mode. This is workload dependent.
- Host Ethernet Adapters require 40 to 56 MB for memory per logical port to vary on.
- IBM Power 550, 9409-M50 May show 2 to 5 percent increase over IBM Power 520, 9408-M25 due to the incorporation of L3 cache. Results will vary based on workload and configuration.
- Virtual ethernet should always be configured with jumbo frame enabled
- In 6.1 Packet Trainer is defaulted to "off" but can be configured per Line Description in 6.1.
- Virtual ethernet may see performance increases with Packet Trainer turn on. This depends on workload, connection type and utilization.
- Physical Gigabit lines may see performance increases with Packet Trainer off. This depends on workload, connection type and utilization.
- Host Ethernet Adapter should not be used for performance sensitive workloads, your throughput can be greatly affected by the use of other logical ports connected to your physical port on additional partitions.
- Host Ethernet Adapter may see performance increases with Packet Trainer set to on, especially with regard to HEA's internal Logical Switch and Partition to Partition traffic via the same port group.

- For additional information regarding your Host Ethernet Adapter please see your specification manual and the Performance Management page for future white papers regarding iSeries and HEA.

- 1 Gigabit Jumbo frame Ethernet enables 12% greater throughput compared to normal frame 1 Gigabit Ethernet. This may vary significantly based on your system, network and workload attributes. Measured 1 Gigabit Jumbo Frame Ethernet throughput approached 1 Gigabit/sec

- The jumbo frame option requires 8992 Byte MTU support by all of the network components including switches, routers and bridges. For System Adapter configuration, LINESPEED(*AUTO) and DUPLEX(*FULL) or DUPLEX(*AUTO) must also be specified. To confirm that jumbo frames have been successfully configured throughout the network, use NETSTAT option 3 to "Display Details" for the active jumbo frame network connection.

- Using ***ETHV2** for the "Ethernet Standard" attribute of **CRTLINETH** may see slight performance increase in STREAMING workloads for 1 Gigabit lines.

- Always ensure that the entire communications network is configured optimally. The **maximum frame size parameter** (MAXFRAME on LIND) should be maximized. The **maximum transmission unit (MTU) size** parameter (CFGTCP command) for both the interface and the route affect the actual size of the line flows and should be configured to *LIND and *IFC respectively. Having configured a large frame size does not negatively impact performance for small transfers. Note that both the System i and the other link station must be configured for large frames. Otherwise, the smaller of the two maximum frame size values is used in transferring data. Bridges may also limit the maximum frame size.

- When transferring large amounts of data, maximize the size of the application's send and receive requests. This is the amount of data that the application transfers with a single sockets API call. Because sockets does not block up multiple application sends, it is important to block in the application if possible.

- With the CHGTCPA command using the parameters TCPRCVBUF and TCPSNDBUF you can alter the TCP receive and send buffers. When transferring large amounts of data, you may experience higher throughput by increasing these buffer sizes up to 8MB. The exact buffer size that provides the best throughput will be dependent on several network environment factors including types of switches and systems, ACK timing, error rate and network topology. In our test environment we used 1 MB buffers. Read the help for this command for more information.

- Application time for transfer environments, including accessing a data base file, decreases the maximum potential data rate. Because the CPU has additional work to process, a smaller percentage of the CPU is available to handle the transfer of data. Also, serialization from the application's use of both database and communications will reduce the transfer rates.

- TCP/IP Attributes (CHGTCPA) now includes a parameter to set the TCP closed connection wait time-out value (TCPCLOTIMO) . This value indicates the amount of time, in seconds, for which a socket pair (client IP address and port, server IP address and port) cannot be reused after a connection is closed. Normally it is set to at least twice the maximum segment lifetime. For typical applications the default value of 120 seconds, limiting the system to approximately 500 new socket pairs per second, is fine. Some applications such as primitive communications benchmarks work best if this setting reflects a value closer to twice the true maximum segment lifetime. In these cases a setting of

only a few seconds may perform best. Setting this value too low may result in extra error handling impacting system capacity.

- No single station can or is expected to use the full bandwidth of the LAN media. It offers up to the media's rated speed of aggregate capacity for the attached stations to share. The disk access time is usually the limiting resource. The data rate is governed primarily by the application efficiency attributes (for example, amount of disk accesses, amount of CPU processing of data, application blocking factors, etc.).

- LAN can achieve a significantly higher data rate than most supported WAN protocols. This is due to the desirable combination of having a high media speed along with optimized protocol software.

- Communications applications consume CPU resource (to process data, to support disk I/O, etc.) and communications line resource (to send and receive data). The amount of line resource that is consumed is proportional to the total number of bytes sent or received on the line. Some additional CPU resource is consumed to process the communications software to support the individual sends (puts or writes) and receives (gets or reads).

- When several sessions use a line concurrently, the aggregate data rate may be higher. This is due to the inherent inefficiency of a single session in using the link. In other words, when a single job is executing disk operations or doing non-overlapped CPU processing, the communications link is idle. If several sessions transfer concurrently, then the jobs may be more interleaved and make better use of the communications link.

- The CPU usage for high speed connections is similar to "slower speed" lines running the same type of work. As the speed of a line increases from a traditional low speed to a high speed, performance characteristics may change.
    - Interactive transactions may be slightly faster
    - Large transfers may be significantly faster
    - A single job may be too serialized to utilize the entire bandwidth
    - High throughput is more sensitive to frame size
    - High throughput is more sensitive to application efficiency
    - System utilization from other work has more impact on throughput

- When developing scalable communication applications, consider taking advantage of the Asynchronous and Overlapped I/O Sockets interface. This interface provides methods for threaded client server model applications to perform highly concurrent and have memory efficient I/O. Additional implementation information is available in the Sockets Programming guide.

---

## 5.7 APPC, ICF, CPI-C, and Anynet

- Ensure that APPC is configured optimally for best performance: LANMAXOUT on the CTLD (for APPC environments): This parameter governs how often the sending system waits for an acknowledgment. Never allow LANACKFRQ on one system to have a greater value than LANMAXOUT on the other system. The parameter values of the sending system should match the values on the receiving system. In general, a value of *CALC (i.e., LANMAXOUT=2) offers the best performance for interactive environments, and adequate performance for large transfer environments. For large transfer environments, changing LANMAXOUT to 6 may provide a significant performance increase. LANWNWSTP for APPC on the controller description (CTLD): If

there is network congestion or overruns to certain target system adapters, then increasing the value from the default=*NONE to 2 or something larger may improve performance. MAXLENRU for APPC on the mode description (MODD): If a value of *CALC is selected for the maximum SNA request/response unit (RU) the system will select an efficient size that is compatible with the frame size (on the LIND) that you choose. The newer LAN IOPs support IOP assist. Changing the RU size to a value other than *CALC may negate this performance feature.

- Some APPC APIs provide blocking (e.g., ICF and CPI-C), therefore scenarios that include repetitive small puts (that may be blocked) may achieve much better performance.
- A large transfer with the System i sending each record repetitively using the default blocking provided by OS/400 to the System i client provides the best level of performance.
- A large transfer with the System i flushing the communications buffer after each record (FRCDTA keyword for ICF) to the System i client consumes more CPU time and reduces the potential data rate. That is, each record will be forced out of the server system to the client system without waiting to be blocked with any subsequent data. Note that ICF and CPI-C support blocking, Sockets does not.
- A large transfer with the System i sending each record requiring a synchronous confirm (e.g., CONFIRM keyword for ICF) to the System is client uses even more CPU and places a high level of serialization reducing the data rate. That is, each record is forced out of the server system to the client system. The server system program then waits for the client system to respond with a confirm (acknowledgment). The server application cannot send the next record until the confirm has been received.
- Compression with APPC should be used with caution and only for slower speed WAN environments. Many suggest that compression should be used with speeds 19.2 kbps and slower and is dependent on the data being transmitted (# of blanks, # and type of repetitions, etc.). Compression is very CPU-intensive. For the CPB benchmark, compression increases the CPU time by up to 9 times. RLE compression uses less CPU time than LZ9 compression (MODD parameters).
- ICF and CPI-C have very similar performance for small data transfers.
- ICF allows for locate mode which means one less move of the data. This makes a significant difference when using larger records.
- The best case data rate is to use the normal blocking that OS/400 provides. For best performance, the use of the ICF keywords force data and confirm should be minimized. An application's use of these keywords has its place, but the tradeoff with performance should be considered. Any deviation from using the normal blocking that OS/400 provides may cause additional trips through the communications software and hardware; therefore, it increases both the overall delay and the amount of resources consumed.
- Having ANYNET = *YES causes extra CPU processing. Only have it set to *YES if it is needed functionally; otherwise, leave it set to *NO.
- For send and receive pairs, the most efficient use of an interface is with it's "native" protocol stack. That is, ICF and CPI-C perform the best with APPC, and Sockets performs best with TCP/IP. There is CPU time overhead when the "cross over" is processed. Each interface/stack may perform differently depending on the scenario.
- Copyfile with DDM provides an efficient way to transfer files between System i systems. DDM provides large blocking which limits the number of times the communications support is invoked. It also maximizes efficiencies with the data base by doing fewer larger I/Os. Generally, a higher data rate can be achieved with DDM compared with user-written APPC programs (doing data base accesses) or with ODF.
- When ODF is used with the SNDNETF command, it must first copy the data to the distribution queue on the sending system. This activity is highly CPU-intensive and takes a considerable amount of time. This time is dependent on the number and size of the records in the file. Sending an object to more than one target System i server only requires one copy to the distribution queue. Therefore, the realized data rate may appear higher for the subsequent transfers.

- FTS is a less efficient way to transfer data. However, it offers built in data compression for line speeds less than a given threshold. In some configurations, it will compress data when using LAN; this significantly slows down LAN transfers.

## 5.8 HPR and Enterprise extender considerations

Enterprise Extender is a protocol that allows the transmission of APPC data over IP only infrastructure. In System i support for Enterprise Extender is added in 5.4. The communications using Enterprise Extender protocol can be achieved by creating a special kind of APPC controller, with LINKTYPE parameter of *HPRIP.

Enterprise Extender (*HPRIP) APPC controllers are not attached to a specific line. Because of this, the controller uses the LDLCLNKSPD parameter to determine the initial link speed to the remote system. After a connection has been started, this speed is adjusted automatically, using the measured network values. However if the value of LDLCLNKSPD is too different to the real link speed value at the beginning, the initial connections will not be using optimally the network. A high value will cause too many packets to be dropped, and a low value will cause the system not to reach the real link speed for short bursts of data.

In a laboratory controlled environment with an isolated 100 Mbps Ethernet network, the following average response times were observed on the system (**not** including the time required to start a SNA session and allocate a conversation):

*Table 5.9*

| **Test Type** | **HPRIP Link Speed = 10Mbps** | **HPRIP Link Speed = 100Mbps** | **AnyNet** | **LAN** |
|---|---|---|---|---|
| Short Request with echo | 0.001 sec | 0.001 sec | 0.001 sec | 0.001 sec |
| Short Request | 0.001 sec | 0.001 sec | 0.003 sec | 0.003 sec |
| 64K Request with echo | 0.019 sec | 0.010 sec | 13 sec | 2 sec |
| 64K Request | 0.019 sec | 0.010 sec | 5 sec | 1 sec |
| 1GB Request with echo | 6:14 min | 6:08 min | 7:22 min | 6:04 min |
| 1GB Request | 2:32 min | 2:17 min | 3:33 min | 3:00 min |
| Send File using sndnetf (1GB) | 5:12 min | 5:16 min | 5:40 min | 5:23 min |

The tests were done between two IBM System i5 (9406-820 and 9402-400) servers in an isolated network.

Allocation time refers to the time that it takes for the system to start a conversation to the remote system. The allocation time might be greater when a SNA session has not yet started to the remote system. Measured allocation speed times where of 14 ms, in HPRIP systems in average, while in AnyNet allocation times where of 41 ms in average.

The HPRIP controllers have slightly higher CPU usage than controllers that use a direct LAN attach. The CPU usage is similar to the one measured on AnyNet APPC controllers. On laboratory testing, a LAN transaction took 3 CPW, while HPRIP and AnyNet, both took 3.7 CPW.

## 5.9 Additional Information

Extensive information can be found at the System i Information Center web site at: http://www.ibm.com/eserver/iseries/infocenter .

- For network information select *"Networking"*:
  - See *"TCP/IP setup" → "Internet Protocol version 6"* for IPv6 information
  - See *"Network communications" → "Ethernet"* for Ethernet information.
- For application development select *"Programming"*:
  - See *"Communications" → "Socket Programming"* for the Sockets Programming guide.

Information about Ethernet cards can be found at the IBM Systems Hardware Information Center. The link for this information center is located on the IBM Systems Information Centers Page at: http://publib.boulder.ibm.com/eserver .

- See *"Managing your server and devices" → "Managing devices" → "Managing Peripheral Component Interconnect (PCI) adapters"* for Ethernet PCI adapters information.

# Chapter 6. Web Server and WebSphere Performance

This section discusses System i performance information in Web serving and WebSphere environments. Specific products that are discussed include: HTTP Server (powered by Apache) (in section 6.1), PHP - Zend Core for i (6.2), WebSphere Application Server and WebSphere Application Server - Express (6.3), Web Facing (6.4), Host Access Transformation Services (6.5), System Application Server Instance (6.6), WebSphere Portal Server (6.7), WebSphere Commerce (6.8), WebSphere Commerce Payments (6.9), and Connect for iSeries (6.10).

The primary focus of this section will be to discuss the performance characteristics of the System i platform as a server in a Web environment, provide capacity planning information, and recommend actions to help achieve high performance. Having a high-performance network infrastructure is very important for Web environments; please refer to Chapter 5, "Communications Performance" for related information and tuning tips.

**Web Overview:** There are many factors that can impact overall performance (e.g., end-user response time, throughput) in the complex Web environment, some of which are listed below:

1) Web Browser or client
- processing speed of the client system
- performance characteristics and configuration of the Web browser
- client application performance characteristics

2) Network
- speed of the communications links
- capacity and caching characteristics of any proxy servers
- the responsiveness of any other related remote servers (e.g., payment gateways)
- congestion of network resources

3) System i Web Server and Applications
- System i processor capacity (indicated by the CPW value)
- utilization of key System i server resources (CPU, IOP, memory, disk)
- Web server performance characteristics
- application (e.g., CGI, servlet) performance characteristics

**Comparing traditional communications to Web-based transactions:** For commercial applications, data accesses across the Internet differs distinctly from accesses across 'traditional' communications networks. The additional resources to support Internet transactions by the CPU, IOP, and line are significant and must be considered in capacity planning. Typically, in a traditional network:
- there is a request and response (between client and server)
- connections/sessions are maintained between transactions
- networks are well-understood and tuned

Typically for Web transactions, there may be a dozen or more line transmissions per transaction:
- a connection is established/closed for each transaction
- there is a request and response (between client and server)
- one user transaction may contain many separate Internet transactions
- secure transactions are more frequent and consume more resource
- with the Internet, the network may not be well-understood (route, components, performance)

**Information source and disclaimer:** The information in the sections that follow is based on performance measurements and analysis done in the internal IBM performance lab. The raw data is not provided here, but the highlights, general conclusions, and recommendations are included. Results listed here do not represent any particular customer environment. Actual performance may vary significantly from what is provided here. Note that these workloads are measured in best-case environments (e.g., local LAN, large MTU sizes, no errors). Real Internet networks typically have higher contention, higher levels of logging and security, MTU size limitations, and intermediate network servers (e.g., proxy, SOCKS); and therefore, it would likely consume more resources.

---

## 6.1 HTTP Server (powered by Apache)

The HTTP Server (powered by Apache) for i5/OS has some exciting new features for V5R4. The level of the HTTP Server has been increased to support Apache 2.0.52 and is now a UTF-8 server. This means that requests are being received and then processed as UTF-8 rather than first being converted to EBCDIC and then processed. This will make porting open source modules for the HTTP Server on your IBM System i easier than before. For more information on what's new for HTTP Server for i5/OS, visit http://www.ibm.com/servers/eserver/iseries/software/http/news/sitenews.html

This section discusses some basic information about HTTP Server (powered by Apache) and gives you some insight about the relative performance between primitive HTTP Server tests.

**The typical high-level flow for Web transactions:** the connection is made, the request is received and processed by the HTTP server, the response is sent to the browser, and the connection is ended. If the browser has multiple file requests for the same HTTP server, it is possible to get the multiple requests with one connection. This feature is known as *persistent connection* and can be set using the KeepAlive directive in the HTTP server configuration.

To understand the test environment and to better interpret performance tools reports or screens it is helpful to know that the following jobs and tasks are involved: communications router tasks (IPRTRnnn), several HTTP jobs with at least one with many threads, and perhaps an additional set of application jobs/threads.

**"Web Server Primitives" Workload Description**: The "Web Server Primitives" workload is driven by the program ApacheBench 2.0.40-dev that runs on a client system and simulates multiple Web browser clients by issuing URL requests to the Web Server. The number of simulated clients can be adjusted to vary the offered load, which was kept at a moderate level. Files and programs exist on the IBM System i platform to support the various transaction types. Each of the transaction types used are quite simple, and will serve a static response page of specified data length back to the client. Each of the transactions can be served in a secure (HTTPS:) or a non-secure (HTTP:) fashion. The HTTP server environment is a partition of an IBM System i 570+ 8-Way (2.2Ghz), configured with one dedicated CPU and a 1 Gbps communication adapter.

- **Static Page:** HTTP retrieves a file from IFS and serves the static page. The HTTP server can be configured to cache the file in its local cache to reduce server resource consumption. FRCA (Fast Response Caching Accelerator) can also be configured to cache the file deeper in the operating system and further reduce resource consumption.

- **CGI**: HTTP invokes a CGI program which builds a simple HTML page and serves it via the HTTP server. This CGI program can run in either a new or a named activation group. The CGI programs were compiled using a "named" activation group unless specified otherwise.

**Web Server Capacity Planning**: Please use the IBM Systems Workload Estimator to do capacity planning for Web environments using the following workloads: Web Serving, WebSphere, WebFacing, WebSphere Portal Server, WebSphere Commerce. This tool allows you to suggest a transaction rate and to further characterize your workload. You'll find the tool along with good help text at: http://www.ibm.com/systems/support/tools/estimator . Work with your marketing representative to utilize this tool (also chapter 23).

The following tables provide a summary of the measured performance data for both static and dynamic Web server transactions. These charts should be used in conjunction with the rest of the information in this section for correct interpretation. Results listed here do not represent any particular customer environment. Actual performance may vary significantly from what is provided here.

**Relative Performance Metrics:**

- *"Relative Capacity Metric:* This metric is used throughout this section to demonstrate the relative capacity performance between primitive tests. Because of the diversity of each environment the ability to scale these results could be challenging, but they are provided to give you an insight into the relation between the performance of each primitive HTTP Server test..

*Table 6.1 i5/OS V5R4 Web Serving Relative Capacity - Static Page*

| Transaction Type: | Relative Capacity Metrics | |
|---|---|---|
| | **Non-secure** | **Secure** |
| **Static Page - IFS** | 2.016 | 1.481 |
| **Static Page - Local Cache** | 3.538 | 2.235 |
| **Static Page - FRCA** | 34.730 | n/a |

**Notes/Disclaimers**:
- Data assumes no access logging, no name server interactions, KeepAlive on, LiveLocalCache off
- Secure: 128-bit RC4 symmetric cipher and MD5 message digest with 1024-bit RSA public/private keys
- These results are relative to each other and do not scale with other environments
- Transactions using more complex programs or serving larger files will have lower capacities that what is listed here.

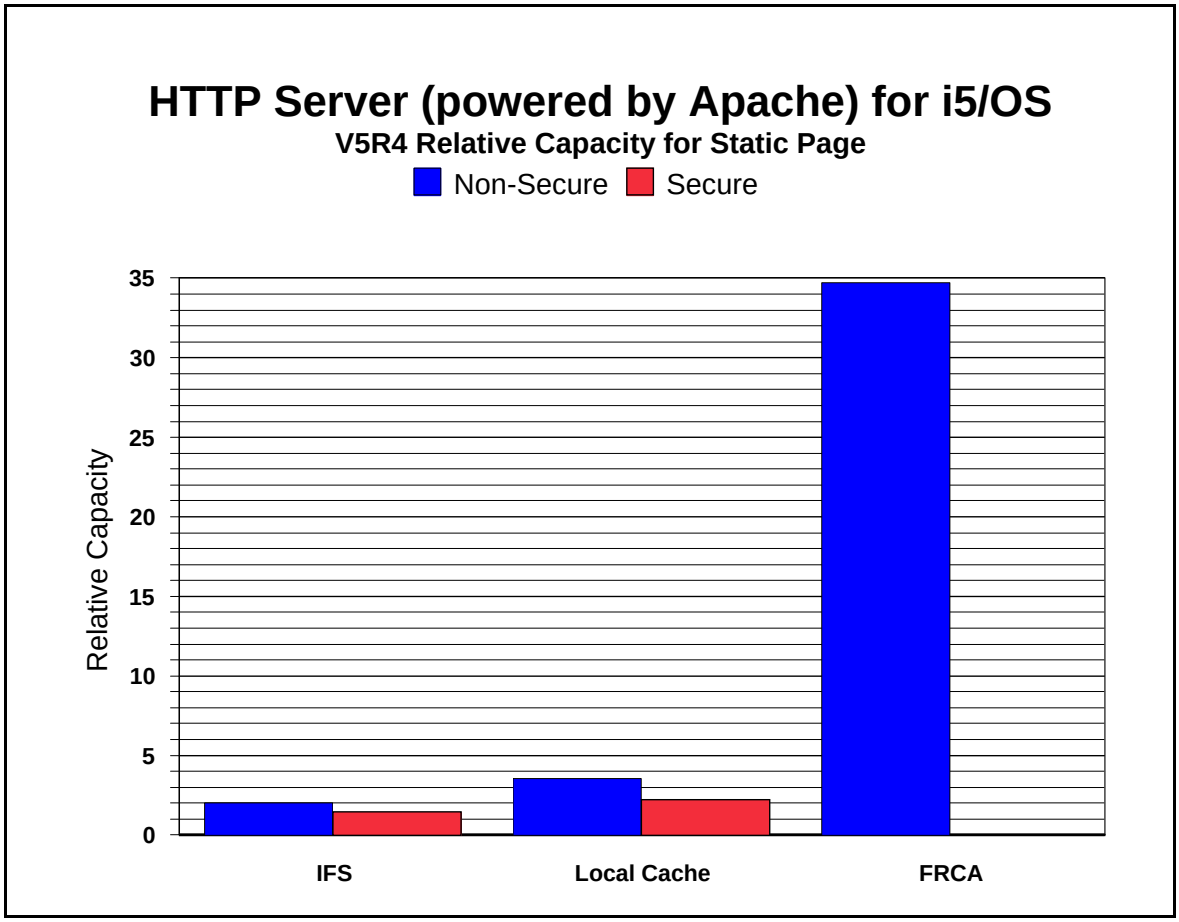



*Figure 6.1 i5/OS V5R4 Web Serving Relative Capacities - Various Transactions*

*Table 6.2 i5/OS V5R4 Web Serving Relative Capacity - CGI*

| Transaction Type: | Relative Capacity Metrics | |
|---|---|---|
| | **Non-secure** | **Secure** |
| **CGI - New Activation** | 0.092 | 0.090 |
| **CGI - Named Activation** | 0.475 | 0.436 |

**Notes/Disclaimers**:
- Data assumes no access logging, no name server interactions, KeepAlive on, LiveLocalCache off
- Secure: 128-bit RC4 symmetric cipher and MD5 message digest with 1024-bit RSA public/private keys
- These results are relative to each other and do not scale with other environments
- Transactions using more complex programs or serving larger files will have lower capacities that what is listed here.

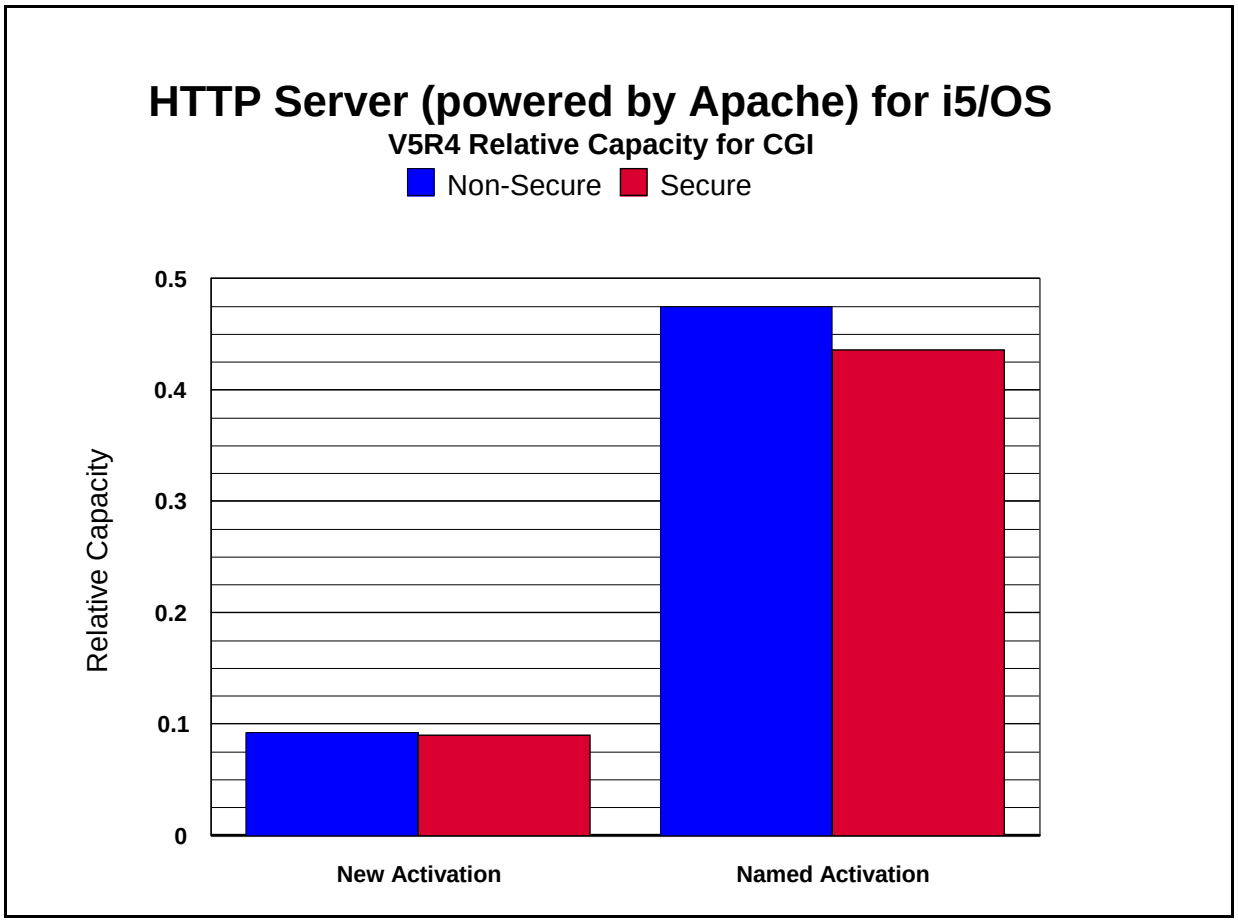



*Figure 6.2 i5/OS V5R4 Web Serving Relative Capacities - Various Transactions*

*Table 6.3 i5/OS V5R4 Web Serving Relative Capacity for Static (varied sizes)*

| | Relative Capacity Metrics | | | | | |
|---|---|---|---|---|---|---|
| **Transaction Type:** | **1K Bytes** | | **`10K Bytes** | | **100K Bytes** | |
| **KeepAlive** | **Off** | **On** | **Off** | **On** | **Off** | **On** |
| **Static Page - IFS** | 1.558 | 2.016 | 1.347 | 1.793 | 0.830 | 1.068 |
| **Static Page - Local Cache** | 2.407 | 3.538 | 2.095 | 3.044 | 0.958 | 1.243 |
| **Static Page - FRCA** | 11.564 | 34.730 | 7.691 | 13.539 | 1.873 | 2.622 |

**Notes/Disclaimers**:
- These results are relative to each other and do not scale with other environments.
- IBM System i CPU features without an L2 cache will have lower web server capacities than the CPW value would indicate

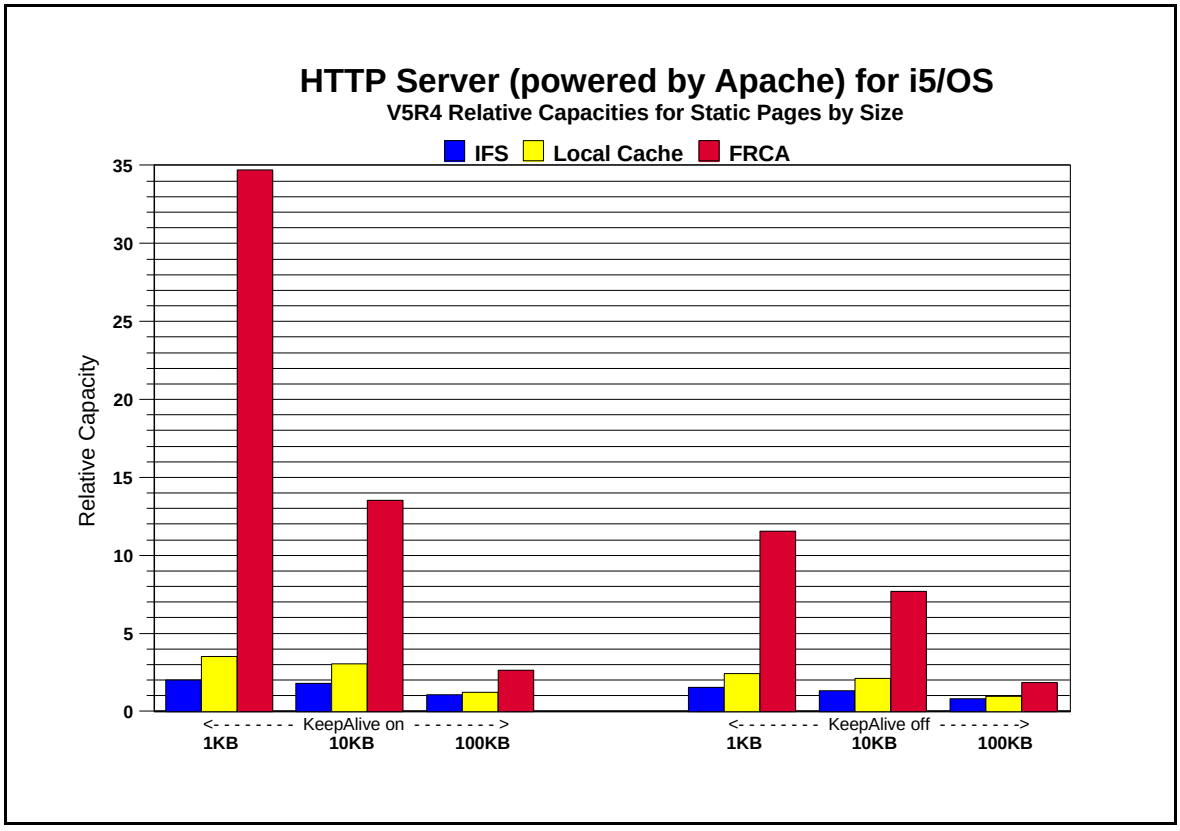



*Figure 6.3 i5/OS V5R4 Web Serving Relative Capacity for Static Pages and FRCA*

### Web Serving Performance Tips and Techniques:

1. **HTTP software optimizations by release:**

a. **V5R4** provides similar Web server performance compared with V5R3 for most transactions (with similar hardware). In V5R4 there are opportunities to exploit improved CGI performance. More information can be found in the FAQ section of the HTTP server website http://www.ibm.com/servers/eserver/iseries/software/http/services/faq.html under "How can I improve the performance of my CGI program?"

b. **V5R3** provided similar Web server performance compared with V5R2 for most transactions (with similar hardware).

c. **V5R2** provided opportunities to exploit improved performance. HTTP Server (powered by Apache) was updated to current levels with improved performance and scalability. FRCA (Fast Response Caching Accelerator) was new with V5R2 and provided a high-performance compliment to the HTTP Server for highly-used static content. FRCA generally reduces the CPU consumption to serve static pages by half, potentially doubling the Web server capacity.

2. **Web Server Cache for IFS Files**: Serving static pages that are cached locally in the HTTP Server's cache can significantly increase Web server capacity (refer to Table 6.3 and Figure 6.3). Ensure that highly used files are selected to be in the cache to limit the overhead of accessing IFS. To keep the cache most useful, it may be best not to consume the cache with extremely large files. Ensure that highly used small/medium files are cached. Also, consider using the LiveLocalCache off directive if possible. If the files you are caching do not change, you can avoid the processing associated with checking each file for any updates to the data. A great deal of caution is recommeded before enabling this directive**.**

3. **FRCA:** Fast Response Caching Accelerator is newly implemented for V5R2. FRCA is based on AFPA (Adaptive Fast Path Architecture), utilizes NFC (Network File Cache) to cache files, and interacts closely with the HTTP Server (powered by Apache). FRCA greatly improves Web server performance for serving static content (refer to Table 6.3 and Figure 6.3). For best performance, FRCA should be used to store static, non-secure content (pages, gifs, images, thumbnails). Keep in mind that HTTP requests served by FRCA are not authenticated and that the files served by FRCA need to have an ASCII CCSID and correct authority. Taking advantage of all levels of caching is really the key for good e-Commerce performance (local HTTP cache, FRCA cache, WebSphere Commerce cache, etc.).

4. **Page size**: The data in the Table 6.1 and Table 6.2 assumes that a small amount of data is being served (say 100 bytes). Table 6.3 illustrates the impact of serving larger files. If the pages are larger, more bytes are processed, CPU processing per transaction significantly increases, and therefore the transaction capacity metrics are reduced. This also increases the communication throughput, which can be a limiting factor for the larger files. The IBM Systems Workload Estimator can be used for capacity planning with page size variations (see chapter 23).

5. **CGI with named activations**: Significant performance benefits can be realized by compiling a CGI program into a "named" versus a "new" activation group, perhaps up to 5x better. It is essential for good performance that CGI-based applications use named activation groups. Refer to the i5/OS ILE Concepts for more details on activation groups. When changing architectures, recompiling CGI programs could boost server performance by taking advantage of compiler optimizations.

6. **Secure Web Serving**: Secure Web serving involves additional overhead to the server for Web environments. There are primarily two groups of overhead: First, there is the fixed overhead of establishing/closing a secure connection, which is dominated by key processing. Second, there is the

variable overhead of encryption/decryption, which is proportional to the number of bytes in the transaction. Note the capacity factors in the tables above comparing non-secure and secure serving. From Table 6.1, note that simple transactions (e.g., static page serving), the impact of secure serving is around 20%. For complex transactions (e.g., CGI, servlets), the overhead is more watered down. This relationship assumes that KeepAlive is used, and therefore the overhead of key processing can be minimized. If KeepAlive is not used (i.e., a new connection, a new cached or abbreviated handshake, more key processing, etc.), then there will be a hit of 7x or more CPU time for using secure transaction. To illustrate this, a noncached SSL static transaction using KeepAlive has a relative capacity of 1.481(from Table 6.1); this compares to 0.188 (not included in the table) when KeepAlive is off. However, if the handshake is forced to be a regular or full handshake, then the CPU time hit will be around 50x (relative capacity 0.03). The lesson here is to: 1) limit the use of security to where it is needed, and 2) use KeepAlive if possible.

7. **Persistent Requests and KeepAlive**: Keeping the TCP/IP connection active during a series of transactions is called persistent connection. Taking advantage of the persistent connection for a series of Web transactions is called Persistent Requests or KeepAlive. This is tuned to satisfy an entire typical Web page being able to serve all imbedded files on that same connection.

    a. Performance Advantages: The CPU and network overhead of establishing and closing a connection is very significant, especially for secure transactions. Utilizing the same connection for several transactions usually allows for significantly better performance, in terms of reduced resource consumption, higher potential capacity, and lower response time.

    b. The down side: If persistent requests are used, the Web server thread associated with that series of requests is tied up (only if the Web Server directive AsyncIO is turned Off). If there is a shortage of available threads, some clients may wait for a thread non-proportionally long. A time-out parameter is used to enforce a maximum amount of time that the connection and thread can remain active.

8. **Logging:** Logging (e.g., access logging) consumes additional CPU and disk resources. Typically, it may consume 10% additional CPU. For best performance, turn off unnecessary logging.

9. **Proxy Servers:** Proxy servers can be used to cache highly-used files. This is a great performance advantage to the HTTP server (the originating server) by reducing the number of requests that it must serve. In this case, an HTTP server would typically be front-ended by one or more proxy servers. If the file is resident in the proxy cache and has not expired, it is served by the proxy server, and the back-end HTTP server is not impacted at all. If the file is not cached or if it has expired, then a request is made to the HTTP server, and served by the proxy.

10. **Response Time (general):** User response time is made up of Web browser (client work station) time, network time, and server time. A problem in any one of these areas may cause a significant performance problem for an end-user. To an end-user, it may seem apparent that any performance problem would be attributable to the server, even though the problem may lie elsewhere. It is common for pages that are being served to have imbedded files (e.g., gifs, images, buttons). Each of these transactions may be a separate Internet transaction. Each adds to the response time since they are treated as independent HTTP requests and can be retrieved from various servers (some browsers can retrieve multiple URLs concurrently). Using Persistent Connection or KeepAlive directive can improve this.

11. **HTTP and TCP/IP Configuration Tips:** Information to assist with the configuration for TCP/IP and HTTP can be viewed at http://publib.boulder.ibm.com/infocenter/iseries/v5r4/index.jsp and http://www.ibm.com/servers/eserver/iseries/software/http/

    a. The **number of HTTP server threads**: The reason for having multiple server threads is that when one server is waiting for a disk or communications I/O to complete, a different server job can process another user's request. Also, if persistent requests are being used and AsyncIO is Off, a server thread is allocated to that user for the entire length of the connection. For N-way systems, each CPU may simultaneously process server jobs. The system will adjust the number of servers that are needed automatically (within the bounds of the minimum and maximum parameters). The values specified are for the number of "worker" threads. Typically, the default values will provide the best performance for most systems. For larger systems, the maximum number of server threads may have to be increased. A starting point for the maximum number of threads can be the CPW value (the portion that is being used for Web server activity) divided by 20. Try not to have excessively more than what is needed as this may cause unnecessary system activity.

    b. The **maximum frame size parameter** (MAXFRAME on LIND) is generally satisfactory for Ethernet because the default value is equal to the maximum value (1.5K). For Token-Ring, it can be increased from 1994 bytes to its maximum of 16393 to allow for larger transmissions.

    c. The **maximum transmission unit (MTU) size** parameter (CFGTCP command) for both the route and interface affect the actual size of the line flows. Optimizing the MTU value will most likely reduce the overall number of transmissions, and therefore, increase the potential capacity of the CPU and the IOP. The MTU on the interface should be set to the frame size (*LIND). The MTU on the route should be set to the interface (*IFC). Similar parameters also exist on the Web browsers. The negotiated value will be the minimum of the server and browser (and perhaps any bridges/routers), so increase them all.

    d. Increasing the **TCP/IP buffer size** (TCPRCVBUF and TCPSNDBUF on the CHGTCPA or CFGTCP command) from 8K bytes to 64K bytes (or as high as 8MB) may increase the performance when sending larger amounts of data. If most of the files being served are 10K bytes or less, it is recommended that the buffer size is not increased to the max of 8MB because it may cause a negative effect on throughput.

    e. **Error and Access Logging**: Having logging turned on causes a small amount of system overhead (CPU time, extra I/O). Typically, it may increase the CPU load by 5-10%. Turn logging off for best capacity. Use the Administration GUI to make changes to the type and amount of logging needed.

    f. **Name Server Accesses**: For each Internet transaction, the server accesses the name server for information (IP address and name translations). These accesses cause significant overhead (CPU time, comm I/O) and greatly reduce system capacity. These accesses can be eliminated by editing the server's config file and adding the line: "HostNameLookups Off".

12. **HTTP Server Memory Requirements**: Follow the faulting threshold guidelines suggested in the work management guide by observing/adjusting the memory in both the machine pool and the pool that the HTTP servers run in (WRKSYSSTS). Factors that may significantly affect the memory requirements include using larger document sizes and using CGI programs**.**

13. **File System Considerations**: Web serving performance varies significantly based on which file system is used. Each file system has different overheads and performance characteristics. Note that serving from the ROOT or QOPENSYS directories provide the best system capacity. If Web page development is done from another directory, consider copying the data to a higher-performing file system for production use. The Web serving performance of the non-thread-safe file systems is significantly less than the root directory. Using QDLS or QSYS may decrease capacity by 2-5 times. Also, be sensitive to the number of sub-directories. Additional overhead is introduced with each sub-directory you add due to the authorization checking that is performed. The HTTP Server serves the pages in ASCII, so make sure that the files have the correct format, else the HTTP Server needs to convert the pages which will result in additional overhead.

14. **Communications/LAN IOPs**: Since there are a dozen or more line flows per transaction (assuming KeepAlive is off), the Web serving environment utilizes the IOP more than other communications environments. Use the Performance Monitor or Collection Services to measure IOP utilization. Attempt to keep the average IOP utilization at 60% or less for best performance. IOP capacity depends on page size, the MTU size, the use of KeepAlive directive, etc. For the best projection of IOP capacity, consider a measurement and observe the IOP utilization.

## 6.2 PHP - Zend Core for i

This section discusses the different performance aspects of running PHP transaction based applications using Zend Core for i, including DB access considerations, utilization of RPG program call, and the benefits of using Zend Platform.

### Zend Core for i

Zend Core for i delivers a rapid development and production PHP foundation for applications using PHP running on i with IBM DB2 for i or MySQL databases. Zend Core for i includes the capability for Web servers to communicate with DB2 and MySQL databases. It is easy to install, and is bundled with Apache 2, PHP 5, and PHP extensions such as ibm_db2.

The PHP application used for this study is a DVD store application that simulates users logging into an online catalog, browsing the catalog, and making DVD purchases. The entire system configuration is a two-tier model with tier one executing the driver that emulates the activities of Web users. Tier two comprises the Web application server that intercepts the requests and sends database transactions to a DB2 for i or MySQL server, configured on the same machine.

### System Configuration

The hardware setup used for this study comprised a driver machine, and a separate system that hosted both the web and database server. The driver machine emulated Web users of an online DVD store generating HTTP requests. These HTTP requests were routed to the Web server that contained the DVD store application logic. The Web server processed the HTTP requests from the Web browsers and maintained persistent connections to the database server jobs. This allowed the connection handle to be preserved after the transaction completed; future incoming transactions re-use the same connection handle. The web and database server was a 2 processor partition on an IBM System i Model 9406-570 server (POWER5 2.2 Ghz) with 2GB of storage. Both IBM i 5.4 and 6.1 were used in the measurements, but for this workload there was minimal difference between the two versions.

### Database and Workload Description

The workload used simulates an Online Transaction Processing (OLTP) environment. A driver simulates users logging in and browsing the catalog of available products via simple search queries. Returning customers are presented with their online purchase transactions history, while new users may register to create customer accounts. Users may select items they would like to purchase and proceed to check out or continue to view available products. In this workload, the browse-buy ratio is 5:1. In total, for a given order (business transaction) there are 10 web requests consisting of login, initiate shopping, five product browse requests, shopping cart update, checkout, and product purchase. This is a transaction oriented workload, utilizing commit processing to insure data integrity. In up to 2% of the orders, rollbacks occur due to insufficient product quantities. Restocking is done once every 30 seconds to replenish the product quantities to control the number of rollbacks.

### Performance Characterization

The metrics used to characterize the performance of the workload were the following:

- Throughput - Orders Per Minute (OPM). Each order actually consists of 10 web requests to complete the order.
- Order response time (RT) in milliseconds
- Total CPU - Total system processor utilization
- CPU Zend/AP - CPU for the Zend Core / Apache component.
- CPU DB - CPU for the DB component

**Database Access**

The following four methods were used to access the backend database for the DVD Store application. In the first three cases, SQL requests were issued directly from the PHP pages. In the fourth case, the i5 PHP API toolkit program call interface was used to call RPG programs to issue i5 native DB IO. For all the environments, the same presentation logic was used.

- ibm_db2 extension shipped with Zend Core for i that provides the SQL interface to DB2 for i.
- mysqli extension that provides the SQL interface to MySQL databases. In this case the MySQL InnoDB and MyISAM storage engines were used.
- i5 PHP API Toolkit SQL functions included with Zend Core for i that provide an SQL interface to DB2 for i.
- i5 PHP API Toolkit classes included with Zend Core for i that provide a program call interface.

When using ibm_db2, there are two ways to connect to DB2. If empty strings are passed for userid and password on the connect, the database access occurs within the same job that the PHP script is executing in. If a specific userid and password are used, database access occurs via a QSQSRVR job, which is called server mode processing. In all tests using ibm_db2, server mode processing was used. This may have a minimal performance impact due to management of QSQSRVR jobs, but does prevent the apache job servicing the php request from not responding if a DB error occurs.

When using ibm_db2 and the i5 toolkit (SQL functions), the accepted practice of using prepare and execute was utilized. In addition stored procedures were utilized for processing the purchase transactions. For MySQL, prepared statements were not utilized because of performance overhead.

Finally, in the case of the i5 PHP API toolkit and ibm_db2, persistent connections were used. Persistent connections provides dramatic performance gains versus using non-persistent connections. This is discussed in more detail in the next section.

In the following table, we compare the performance of the different DB access methods.

| OS / DB | i 5.4 / DB2 | i 5.4 / MySQL 5.0 | i 5.4 / DB2 | i 5.4 / DB2 |
|---|---|---|---|---|
| ZendCore Version | V2.5.2 | V2.5.2 | V2.5.2 | V2.5.2 |
| Connect | db2_pconnect | mysqli | i5_pconnect | i5_pconnect |
| | | | SQL function | Pgm Call Function |
| | | | | |
| OPM | 4997 | 3935 | 3920 | 5240 |
| RT (ms) | 176 | 225 | 227 | 169 |
| Total CPU | 99 | 98 | 99 | 98 |
| CPU - Zend/AP | 62 | 49 | 63 | 88 |
| CPU - DB | 33 | 47 | 33 | 7 |

Conclusions:

1. The performance of each DB connection interface provides exceptional response time at very high throughput. Each order processed consisted of ten web requests. As a result, the capacity ranges from about 650 transactions per second up to about 870 transactions per second. Using Zend Platform will provide even higher performance (refer to the section on Zend Platform).

2. The i5 PHP API Toolkit is networked enabled so provides the capability to run in a 3-tier environment, ie, where the PHP application is running on web server deployed on a separate system from the backend DB server. However, when running in a 2- tier environment, it is recommended to use the ibm_db2 PHP extension to access DB2 locally given the optimized performance.

The i5 PHP API Toolkit provides a wealth of interfaces to integrate PHP pages with native i5 system services. When standardizing on the use of the i5 toolkit API, the use of the SQL functions to access DB2 will provide very good performance. In addition to SQL functions, the toolkit provides a program call interface to call existing programs. Calling existing programs using native DB IO may provide significantly more performance.

3. The most compelling reason to use MySQL on IBM i is when you are deploying an application that is written to the MySQL database.

**Database - Persistent versus Non-Persistent Connections**

If you're connecting to a DB2 database in your PHP application, you'll find that there are two alternative connections - db2_connect which establishes a new connection each time and db2_pconnect which uses persistent connections. The main advantage of using a persistent connection is that it avoids much of the initialization and teardown normally associated with getting a connection to the database. When db2_close() is called against a persistent connection, the call always returns TRUE, but the underlying DB2 client connection remains open and waiting to serve the next matching db2_pconnect() request.

One main area of concern with persistent connections is in the area of commitment control. You need to be very diligent when using persistent connections for transactions that require the use of commitment control boundaries. In this case, DB2_AUTOCOMMIT_OFF is specified and the programmer controls the commit points using db2_commit() statements. If not managed correctly, mixing managed commitment control and persistent connections can result in unknown transaction states if errors occur.

In the following table, we compare the performance of utilizing non-persistent connections in all cases versus using a mix of persistent and non-persistent connections versus using persistent connections in all cases.

| OS / DB | i 5.4 / DB2 | i 5.4 / DB2 | i 5.4 / DB2 |
|---|---|---|---|
| ZendCore Version | V2.5.2 | V2.5.2 | V2.5.2 |
| Connect | db2_connect | Mixed | db2_pconnect |
| | | | |
| OPM | 445 | 2161 | 4997 |
| RT (ms) | 2021 | 414 | 176 |
| Total CPU | 91 | 99 | 99 |
| CPU - Zend/AP | 9 | 33 | 62 |
| CPU - DB | 78 | 62 | 33 |

Conclusions:

1. As stated earlier, persistent connections can dramatically improve overall performance. When using persistent connections for all transactions, the DB CPU utilization is significantly less than when using non-persistent connections.

2. For any transactions that run with autocommit turned on, use persistent connections. If the transaction requires that autocommit be turned off, use of non-persistent connections may be sufficient for pages that don't have heavy usage. However, if a page is heavily used, use of persistent connections may be required to achieve acceptable performance. In this case, you will need a well designed transaction that handles error processing to ensure no commits are left outstanding.

**Database - Isolation Levels**

Because the transaction isolation level determines how data is locked and isolated from other processes while the data is being accessed, you should select an isolation level that balances the requirements of concurrency and data integrity. DB2_I5_TXN_SERIALIZABLE is the most restrictive and protected transaction isolation level, and it incurs significant overhead. Many workloads do not require this level of isolation protection. We did limited testing comparing the performance of using DB2_I5_TXN_READ_COMMITTED versus DB2_I5_TXN_READ_UNCOMMITTED isolation levels. With this workload, running under DB2_I5_TXN_READ_COMMITTED reduced the overall capacity by about 5%. However a given application might never update the underlying data or run with other concurrent updaters and DB2_I5_TXN_READ_UNCOMMITTED may be sufficient. Therefore, review your isolation level requirements and adjust them appropriately.

**Zend Platform**

Zend Platform for i is the production environment that ensures PHP applications are always available, fast, reliable and scalable on the i platform. Zend Platform provides caching and optimization of compiled PHP code, which provides significant performance improvement and scalability. Other features of Zend Platform that brings additional value, include:

- 5020 Bridge – API for accessing 5250 data streams which allows Web front ends to be created for existing applications.
- PHP Intelligence – provides monitoring of PHP applications and captures all the information needed to pinpoint the root cause of problems and performance bottlenecks.
- Online debugging and immediate error resolution with Zend Studio for i
- PHP/Java integration bridge

By automatically caching and optimizing the compiled PHP code, application response time and system capacity improves dramatically. The best part for this is that no changes are required to the take advantage of this optimization. In the measurements included below, the default Zend Platform settings were used.

| OS / DB | i 6.1 / DB2 | | i 6.1/MySQL 5.0 | |
|---|---|---|---|---|
| Zend Version | V2.5.2 | V2.5.2/Platform | V2.5.2 | V2.5.2/Platform |
| Connect | db2_pconnect | db2_pconnect | mysqli | mysqli |
| | | | | |
| OPM | 5041 | 6795 | 3974 | 4610 |
| RT (ms) | 176 | 129 | 224 | 191 |
| Total CPU | 98 | 95 | 98 | 96 |
| CPU - Zend/AP | 62 | 44 | 49 | 31 |
| CPU - DB | 31 | 46 | 47 | 62 |

Conclusions:

1. In both cases above, the overall system capacity improved significantly when using Zend Platform, by about 15-35% for this workload. With each order consisting of 10 web requests, processing 6795 orders per minute translates into about 1132 transactions per second.

2. Zend Platform will reduce the amount of processing in the Zend Core component since the PHP code is compiled once and reused. In both of the above cases, the amount of processing done in Zend Core on a per transaction basis was dramatically reduced by a factor of about 1.9X.

**PHP System Sizing**

The IBM Systems Workload Estimator (a.k.a., the Estimator or WLE) is a web-based sizing tool for IBM Power Systems, System i, System p, and System x. You can use this tool to size a new system, to size an upgrade to an existing system, or to size a consolidation of several systems. The Estimator allows measurement input to best reflect your current workload and provides a variety of built-in workloads to reflect your emerging application requirements.

Currently, a new built-in workload is being developed to allow the sizing of PHP workloads on Power Systems running IBM i. This built-in is expected to be available November 2008. To access WLE use the following URL:

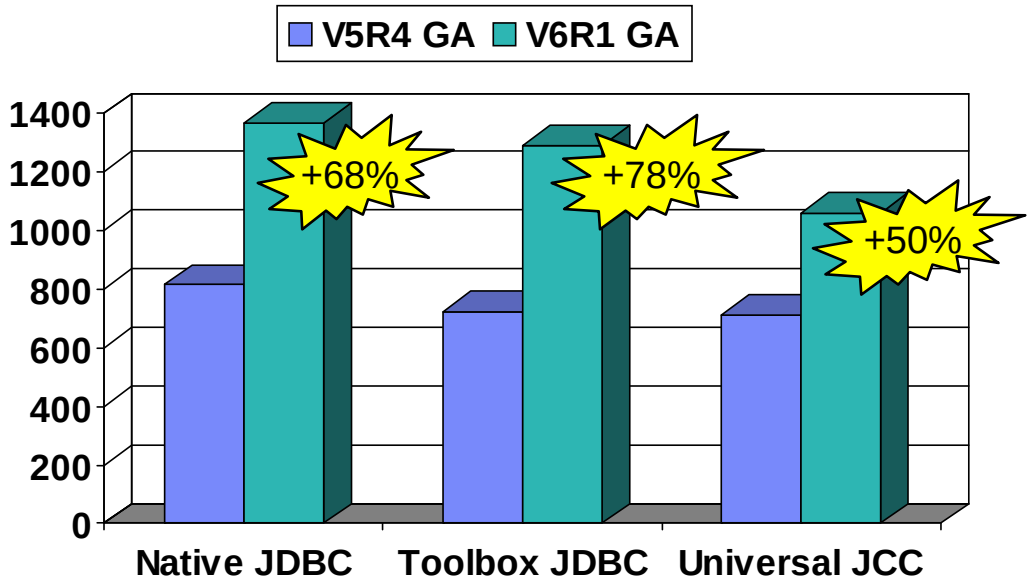
http://www.ibm.com/eserver/iseries/support/estimator

## 6.3 WebSphere Application Server

This section discusses System i performance information for the WebSphere Application Server, including WebSphere Application Server V6.1, WebSphere Application Server V6.0, WebSphere Application Server V5.0 and V5.1, and WebSphere Application Server Express V5.1. Historically, both WebSphere and i5/OS Java performance improve with each version. Note from the figures and data in this section that the most recent versions of WebSphere and/or i5/OS generally provides the best performance.

**What's new in V6R1?**

The release of i5/OS V6R1 brings with it significant performance benefits for many WebSphere applications. The following chart shows the amount of improvement in transactions per second (TPS) for the Trade 6.1 workload using various data access methods:




This chart shows that in V6R1, throughput levels for Trade 6.1 increased from 50% to nearly 80% versus V5R4, depending on which JDBC provider was being used. All measurement results were obtained in a 2-tier environment (both application and database on the same partition) on a 2-core 2.2Ghz System i partition, using Version 6.1 of WebSphere Application Server and IBM Technology for Java VM. Although many of the improvements are applicable to 3-tier environments as well, the communications overhead in these environments may affect the amount of improvement that you will see.

The improvements in V6R1 were primarily in the JDBC, DB2 for i5/OS and Java areas, as well as changes in other i5/OS components such as seize/release and PASE call overhead. The majority of the improvements will be achieved without any changes to your application, although some improvements do require additional tuning (discussed below in **Tuning Changes for V6R1**). Although some of the changes are now available via V5R4 PTFs, the majority of the improvement will only be realized by moving to V6R1. The actual amount of improvement in any particular application will vary, particularly depending on the amount of JDBC/DB activity, where a significant majority of the changes were made. In addition,

because the improvements largely resulted from significant reductions in pathlength and CPU, environments that are constrained by other resources such as IO or memory may not show the same level of improvements seen here.

**Tuning changes in V6R1**

As indicated above, most improvements will require no changes to an application. However, there are a few changes that will require some tuning in order to be realized:

- **Using direct map (native JDBC)**

  For System i, the JDBC interfaces run more efficiently if direct mapping of data is used, where the data being retrieved is in a form that closely matches how the data is stored in the database files. In V6R1, significant enhancements were made in JDBC to allow direct map to be used in more cases. For the toolbox and JCC JDBC drivers, where direct map is the default, there is no change needed to realize these gains. However, for native JDBC, you will need to use the "directMap=true" custom property for the datasource in order to maximize the gain from these changes. For Trade 6.1, measurements show that adding this property results in about a 3-5% improvement in throughput. Note that there is no detrimental effect from using this property, since the JDBC interfaces will only use direct map if it is functionally viable to do so.

- **Use of unix sockets (toolbox JDBC)**

  For toolbox JDBC, the default is to use TCP/IP inet sockets for requests between the application server and the database connections. In V6R1, an enhancement was added to allow the use of unix sockets in a 2-tier toolbox environment (application and database reside on the same partition). Using unix sockets for the Trade 6.1 2-tier workload in V6R1 resulted in about an 8-10% improvement in throughput. However, as the default is still to use inet sockets, you will need to ensure that the class path specified in the JDBC provider is set to use the jt400native.jar file (not the jt400.jar file) in order to use unix sockets. Note that the improvement is applicable only to 2-tier toolbox environments. Inet sockets will continue to be used for all other multiple tier toolbox environments no matter which .jar file is used.
  .

- **Using 'threadUsed=false" custom property (toolbox JDBC)**

  In toolbox JDBC, the default method of operation is to use multiple application server threads for each request to a database connection, with one thread used for sending data to the connection and another thread being used to receive data from the connection. In V6R1, changes were made to allow both the send and receive activity to be done within a single application server thread for each request, thus reducing the overhead associated with the multiple threads. To gain the potential improvement from this change, you will need to specify the "threadUsed=false" custom property in the toolbox datasource, since the default is still to use multiple threads. For the Trade 6.1 workload, use of the this property resulted in about a 10% improvement in throughput.

**Tuning for WebSphere** is important to achieve optimal performance. Please refer to the *WebSphere Application Server for iSeries Performance Considerations or the WebSphere Info Center* documents for more information. These documents describe the performance differences between the different WebSphere Application Server versions on the System i platform. They also contain many performance recommendations for environments using servlets, Java Server Pages (JSPs), and Enterprise Java Beans.

**For WebSphere 5.1 and earlier refer to the Performance Considerations guide at:**
www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/product/PerformanceConsiderations.html
**For WebSphere 5.1, 6.0 and 6.1 please refer to the following page and follow the appropriate link:**
www.ibm.com/software/webservers/appserv/was/library/

Although some capacity planning information is included in these documents, please use the IBM Systems Workload Estimator as the primary tool to size WebSphere environments. The Workload Estimator is kept up to date with the latest capacity planning information available.

**Trade 6 Benchmark (**IBM Trade Performance Benchmark Sample for WebSphere Application Server**)**
**Description:**

Trade 6 is the fourth generation of the WebSphere end-to-end benchmark and performance sample application. The Trade benchmark is designed and developed to cover the significantly expanding programming model and performance technologies associated with WebSphere Application Server. This application provides a real-world workload, enabling performance research and verification test of the Java™ 2 Platform, Enterprise Edition (J2EE™) 1.4 implementation in WebSphere Application Server, including key performance components and features**.**

Overall, the Trade application is primarily used for performance research on a wide range of software components and platforms. This latest revision of Trade builds off of Trade 3, by moving from the J2EE 1.3 programming model to the J2EE 1.4 model that is supported by WebSphere Application Server V6.0. Trade 6 adds `DistributedMap` based data caching in addition to the command bean caching that is used in Trade 3. Otherwise, the implementation and workflow of the Trade application remains unchanged.

Trade 6 also supports the recent DB2® V8.2 and Oracle® 10g databases. The new design of Trade 6 enables performance research on J2EE 1.4 including the new Enterprise JavaBeansTM (EJB™) 2.1 component architecture, message-driven beans, transactions (1-phase, 2-phase commit) and Web services (SOAP, WSDL, JAX-RPC, enterprise Web services). Trade 6 also drives key WebSphere Application Server performance components such as dynamic caching, WebSphere Edge Server, and EJB caching.

**NOTE: Trade 6 is an updated version of Trade 3 which takes advantage of the new JMS messaging support available with WebSphere 6.0. The application itself is essentially the same as Trade 3 so direct comparisons can be made between Trade 6 and Trade 3. However, it is important to note that direct comparisons between Trade2 and Trade3 are NOT valid. As a result of the redesign and additional components that were added to Trade 3, Trade 3 is more complex and is a heavier application than the previous Trade 2 versions.**

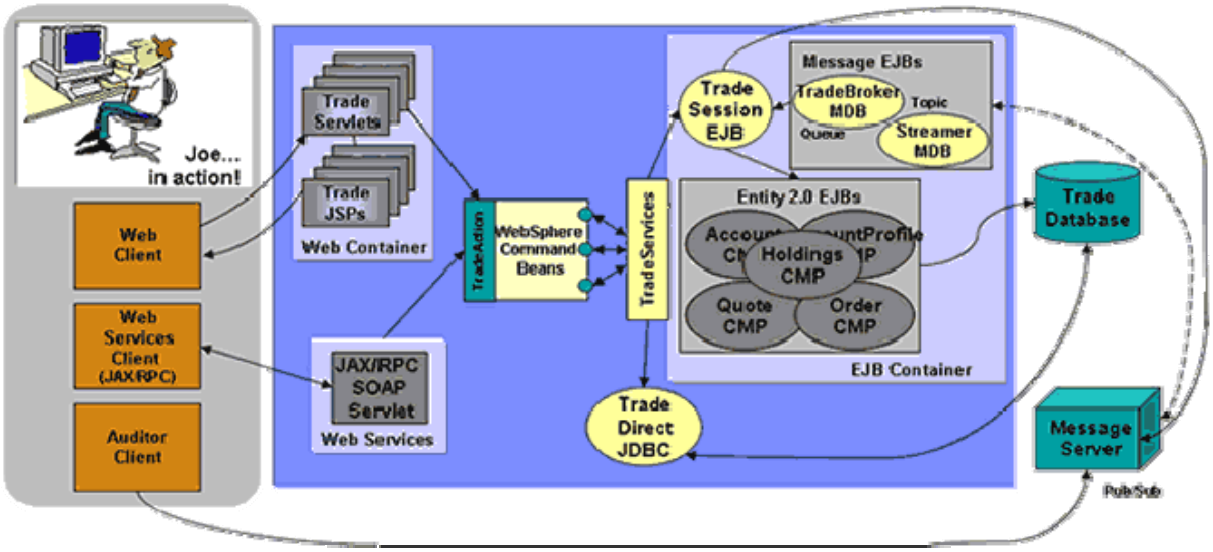



**Figure 6. 1 Topology of the Trade Application**

The Trade 6 application allows a user, typically using a Web browser, to perform the following actions:

- Register to create a user profile, user ID/password and initial account balance
- Login to validate an already registered user
- Browse current stock price for a ticker symbol
- Purchase shares
- Sell shares from holdings
- Browse portfolio
- Logout to terminate the users active interval

Each **action** is comprised of many primitive operations running within the context of a single HTTP request/response. For any given action there is exactly one transaction comprised of 2-5 remote method calls. A ***Sell*** action for example, would involve the following primitive operations:

- Browser issues an HTTP GET command on the TradeAppServlet
- TradeServlet accesses the cookie-based HTTP Session for that user
- HTML form data input is accessed to select the stock to sell
- The stock is sold by invoking the **sell()** method on the **Trade** bean, a stateless ***Session EJB***. To achieve the sell, a transaction is opened and the Trade bean then calls methods on Quote, Account and Holdings ***Entity EJBs*** to execute the sell as a single transaction.
- The results of the transaction, including the new current balance, total sell price and other data, are formatted as HTML output using a Java Server Page, portfolio.jsp.
- Message Driven Beans are used to inform the user that the transaction has completed on the next logon of that user.

To measure performance across various configuration options, the Trade 6 application can be run in several modes. A mode defines the environment and component used in a test and is configured by modifying settings through the Trade 6 interface. For example, data object access can be configured to use JDBC directly or to use EJBs under WebSphere by setting the Trade 6 *runtime mode*. In the ***Sell*** example above, operations are listed for the EJB runtime mode. If the mode is set to JDBC, the *sell* action is completed by direct data access through JDBC from the TradeAppServlet. Several testing modes are available and are varied for individual tests to analyze performance characteristics under various configurations.

## WebSphere Application Server V6.1

Historically, new releases of WebSphere Application Server have offered improved performance and functionality over prior releases of WebSphere. WebSphere Application Server V6.1 is no exception. Furthermore, the availability of WebSphere Application Server V6.1 offers an entirely new opportunity for WebSphere customers. Applications running on V6.1 can now operate with either the "Classic" 64-bit Virtual Machine (VM) or the recently released IBM Technology for Java, a 32-bit VM that is built on technology being introduced across all the IBM Systems platforms.

Customers running releases of WebSphere Application prior to V6.1 will likely be familiar with the Classic 64-bit VM. This continues to be the default VM on i5/OS, offering competitive performance and excellent vertical scalability. Experiments conducted using the Trade6 benchmark show that WebSphere Application Server V6.1 running on the Classic VM realized performance gains of 5-10% better throughput when compared to WebSphere Application Server V6.0 on identical hardware.

In addition to the presence of the Classic 64-bit VM, WebSphere Application Server V6.1 can also take advantage of IBM Technology for Java, a 32-bit implementation of Java supported on Java 5.0 (JDK 1.5). For V6.1 users, IBM Technology for Java has two key potentially beneficial characteristics:

- *Significant performance improvements for many applications* - Most applications will see at least equivalent performance when comparing WebSphere Application Server on the Classic VM to IBM Technology for Java, with many applications seeing improvements of up to 20%.
- *32-bit addressing allows for a potentially considerable reduction in memory footprint* - Object references require only 4 bytes of memory as opposed to the 8 bytes required in the 64-bit Classic VM. For users running on small systems with relatively low memory demands this could offer a substantially smaller memory footprint. Performance tests have shown approximately 40% smaller Java Heap sizes when using IBM Technology for Java when compared to the Classic VM.

It is important to realize that both the Classic VM and IBM Technology for Java have excellent benefits for different applications. Therefore, choosing which VM to use is an extremely important consideration. *Chapter 7 - Java Performance* has an extensive overview of many of the key decisions that go into choosing which VM to use for a given application. Most of the points in Chapter 7 are very much important to WebSphere Application Server users. One issue that will likely not be a concern to WebSphere Application Server users is the additional overhead to native ILE calls that is seen in IBM Technology for Java. However, if native calls are relevant to a particular application, that consideration will of course be important. While choosing the appropriate VM is important, WebSphere Application Server V6.1 allows users to toggle between the Classic VM and IBM Technology for Java either for the entire WebSphere installation or for individual application server profiles.

While 32-bit addressing can provide smaller memory footprints for some applications, it is imperative to understand the other end of the spectrum: applications requiring large Java heaps may not be able to fit in the space available to a 32-bit implementation of Java. The 32-bit VM has a maximum heap size of 3328 MB for Java applications. However, WebSphere Application Server V6.1 using IBM Technology for Java has a practical maximum heap size of around 2500 MB due in part to WebSphere related memory demands like shared classes. The Classic VM should be used for applications that require a heap larger than 2500 MB (see Chapter 7 - Java Performance for further details).

**Trade3 Measurement Results:**

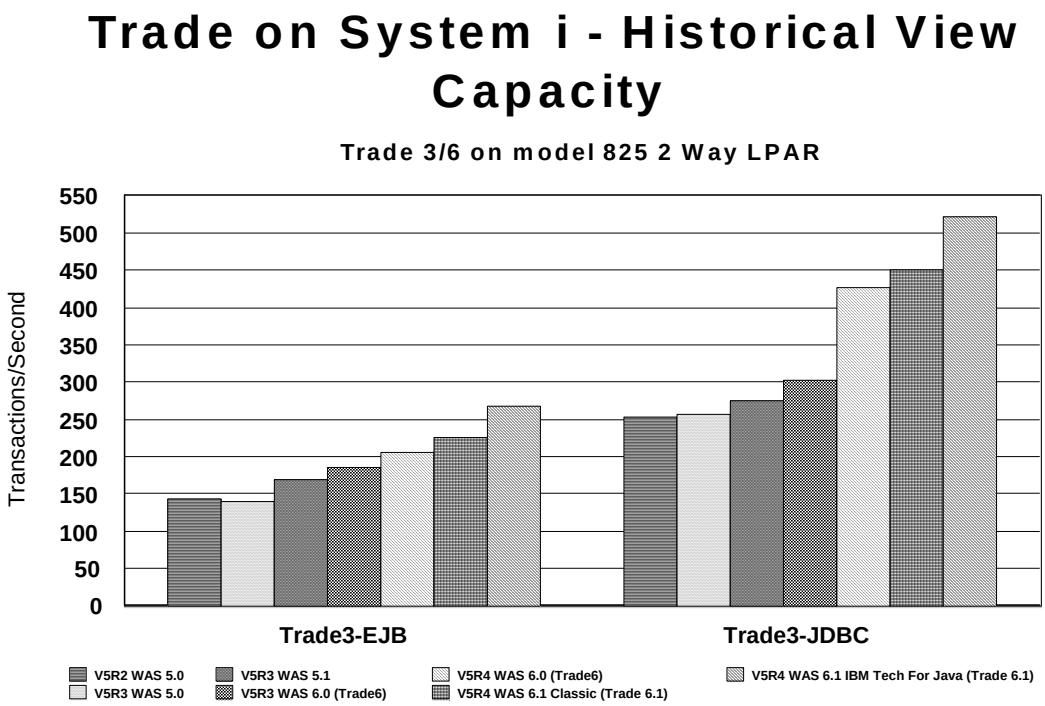



**Figure 6.2 Trade Capacity Results**

| *WebSphere Application Server Trade Results* |
|---|
| **Notes/Disclaimers**: |
| • Trade3 chart:<br>WebSphere 5.0 was measured on both V5R2 and V5R3 on a 4 way (LPAR) 825/2473 system<br>WebSphere 5.1 was measured on V5R3 on a 4 way (LPAR) 825/2473 system<br>WebSphere 6.0 was measured on V5R3 on a 4 way (LPAR) 825/2473 system<br>WebSphere 6.0 was measured on V5R4 on a 2 way (LPAR) 570/7758 system<br>WebSphere 6.1 using Classic VM was measured on V5R4 on a 2 way (LPAR) 570/7758 system<br>WebSphere 6.1 using IBM Technology for Java was measured on V5R4 on a 2 way (LPAR) 570/7758 system |
| |

**Trade Scalability Results**:

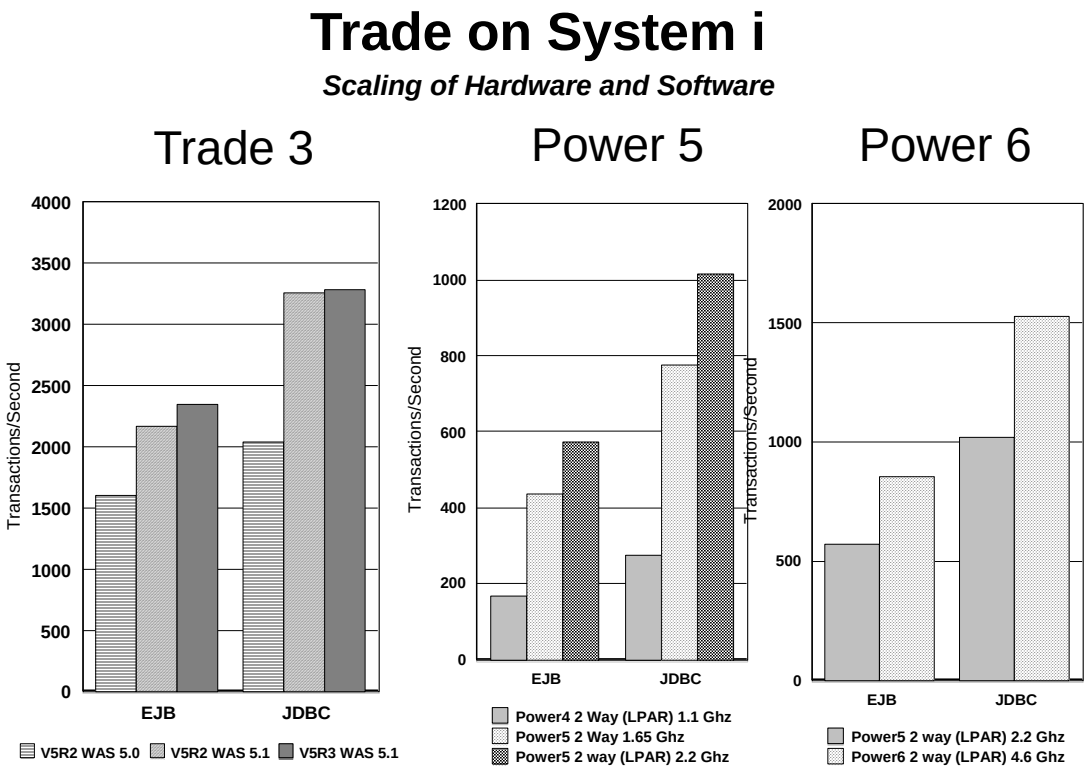



**Figure 6.3 Trade Scaling Results**

| *WebSphere Application Server Trade Results* |
|---|
| **Notes/Disclaimers**: |
| • Trade 3 chart:<br>V5R2 - 890/2488 32-Way 1.3 GHz, V5R2 was measured with WebSphere 5.0 and WebSphere 5.1<br>V5R3 - 890/2488 32-Way 1.3 GHz, V5R3 was measured with WebSphere 5.1<br>POWER5 chart:<br>POWER4 - V5R3 825/2473 2-Way (LPAR) 1.1 GHz., Power4 was measured with WebSphere 5.1<br>POWER5 - V5R3 520/7457 2-Way 1.65 GHz., Power5 was measured with WebSphere 5.1<br>POWER5 - V5R4 570/7758 2-Way (LPAR) 2.2 GHz, Power5 was measured with WebSphere 6.0<br>• POWER6 chart:<br>POWER5 - V5R4 570/7758 2-Way (LPAR) 2.2 GHz, Power5 was measured with WebSphere 6.0<br>POWER6 - V5R4 9406-MMA 2-Way (LPAR) 4.7 GHz, Power6 was measured with WebSphere 6.1 |
| |

### Trade 6 Primitives

Trade 6 provides an expanded suite of Web primitives, which singularly test key operations in the enterprise Java programming model. These primitives are very useful in the Rochester lab for release-to-release comparison tests, to determine if a degradation occurs between releases, and what areas to target performance improvements. Table 6.1 describes all of the primitives that are shipped with Trade 6, and Figure 6.4 shows the results of the primitives from WAS 5.0 and WAS 5.1. In V5R4 a few of the primitives were tracked on WAS 6.0, showing a change of 0-2%, the results of which are not included in Figure 6.4. In the future, additional primitives are planned again to be measured for comparison.

| Primitive Name | Description of Primitive |
|---|---|
| PingHtml | PingHtml is the most basic operation providing access to a simple "Hello World" page of static HTML. |
| PingServlet | PingServlet tests fundamental dynamic HTML creation through server side servlet processing. |
| PingServletWriter | PingServletWriter extends PingServlet by using a PrintWriter for formatted output vs. the output stream used by PingServlet. |
| PingServlet2Include | PingServlet2Include tests response inclusion. Servlet 1 includes the response of Servlet 2. |
| PingServlet2Servlet | PingServlet2Servlet tests request dispatching. Servlet 1, the controller, creates a new JavaBean object forwards the request with the JavaBean added to Servlet 2. Servlet 2 obtains access to the JavaBean through the Servlet request object and provides dynamic HTML output based on the JavaBean data. |
| PingJSP | PingJSP tests a direct call to JavaServer Page providing server-side dynamic HTML through JSP scripting. |
| PingJSPEL | PingJSPEL tests a direct call to JavaServer Page providing server-side dynamic HTML through JSP scripting and the usage of the new JSP 2.0 Expression Language. |
| PingServlet2JSP | PingServlet2JSP tests a commonly used design pattern, where a request is issued to servlet providing server side control processing. The servlet creates a JavaBean object with dynamically set attributes and forwards the bean to the JSP through a RequestDispatcher The JSP obtains access to the JavaBean and provides formatted display with dynamic HTML output based on the JavaBean data. |
| PingHTTPSession1 | PingHTTPSession1 - SessionID tests fundamental HTTP session function by creating a unique session ID for each individual user. The ID is stored in the users session and is accessed and displayed on each user request. |
| PingHTTPSession2 | PingHTTPSession2 session create/destroy further extends the previous test by invalidating the HTTP Session on every 5th user access. This results in testing HTTPSession create and destroy. |
| PingHTTPSession3 | PingHTTPSession3 large session object tests the servers ability to manage and persist large HTTPSession data objects. The servlet creates a large custom java object. The class contains multiple data fields and results in 2048 bytes of data. This large session object is retrieved and stored to the session on each user request. |
| PingJDBCRead | PingJDBCRead tests fundamental servlet to JDBC access to a database performing a single-row read using a prepared SQL statment. |
| PingJDBCWrite | PingJDBCRead tests fundamental servlet to JDBC access to a database performing a single-row write using a prepared SQL statment. |
| PingServlet2JNDI | PingServlet2JNDI tests the fundamental J2EE operation of a servlet allocating a JNDI context and performing a JNDI lookup of a JDBC DataSource. |
| PingServlet2SessionEJB | PingServlet2SessionEJB tests key function of a servlet call to a stateless SessionEJB. The SessionEJB performs a simple calculation and returns the result. |
| PingServlet2EntityEJBLocal<br><br>PingServlet2EntityEJBRemote | PingServlet2EntityEJB tests key function of a servlet call to an EJB 2.0 Container Managed Entity. In this test the EJB entity represents a single row in the database table. The Local version uses the EJB Local interface while the Remote version uses the Remote EJB interface. (Note: PingServlet2EntityEJBLocal will fail in a multi-tier setup where the Trade3 Web and EJB apps are seperated.) |
| PingServlet2Session2Entity | Tests the full servlet to Session EJB to Entity EJB path to retrieve a single row from the database. |
| PingServlet2Session2 EntityCollection | This test extends the previous EJB Entity test by calling a Session EJB which uses a finder method on the Entity that returns a collection of Entity objects. Each object is displayed by the servlet |
| PingServlet2Session2CMROne 2One | This test drives an Entity EJB to get another Entity EJB's data through an EJB 2.0 CMR One to One relationship |
| PingServlet2Session2CMROne 2Many | This test drives an Entity EJB to get another Entity EJB's data through an EJB 2.0 CMR One to Many relationship |
| PingServlet2MDBQueue | PingServlet2MDBQueue drives messages to a Queue based Message Driven EJB (MDB).Each request to the servlet posts a message to the Queue. The MDB receives the message asynchronously and prints message delivery statistics on each 100th message. |
| PingServlet2MDBTopic | PingServlet2MDBTopic drives messages to a Topic based Publish/Subscribe Message Driven EJB (MDB).Each request to the servlet posts a message to the Topic. The TradeStreamMDB receives the message asynchronously and prints message delivery statistics on each 100th message. Other subscribers to the Topic will also receive the messages. |
| PingServlet2TwoPhase | PingServlet2TwoPhase drives a Session EJB which invokes an Entity EJB with findByPrimaryKey (DB Access) followed by posting a message to an MDB through a JMS Queue (Message access). These operations are wrapped in a global 2-phase transaction and commit. |

**Table 6.1** Description of Trade primitives in Figure 6.4

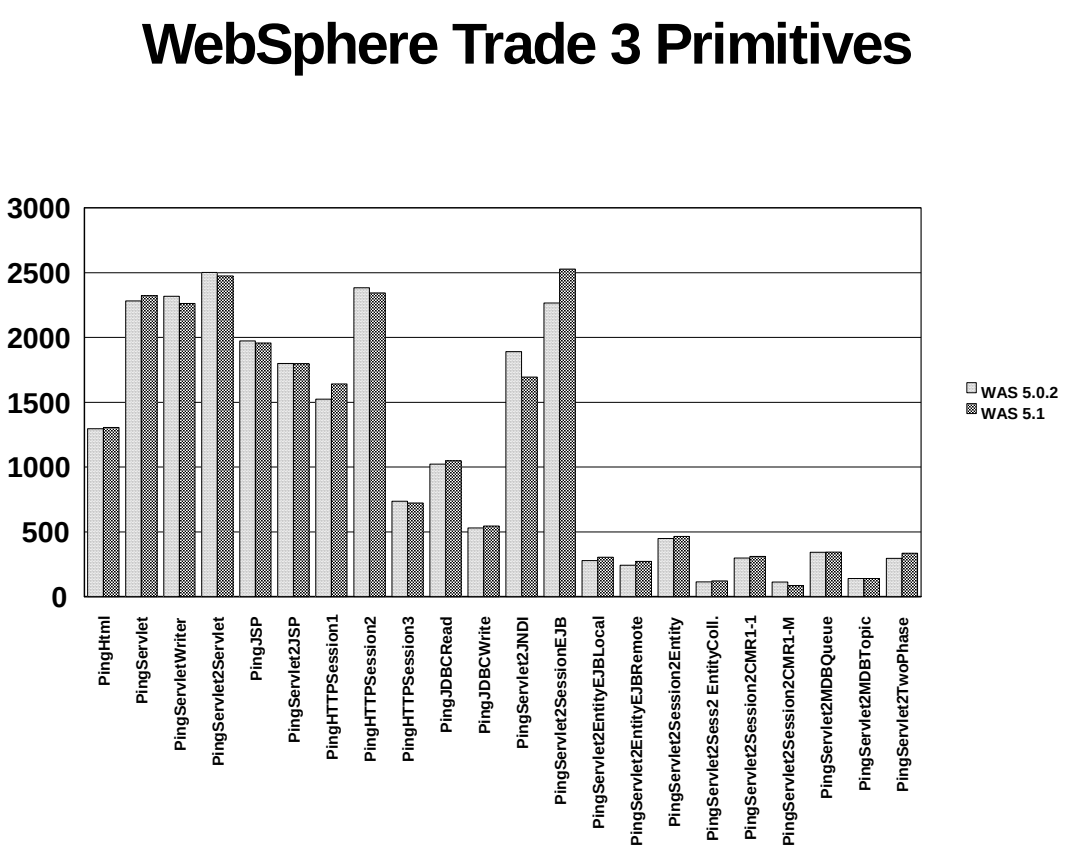



**Figure 6.4 WebSphere Trade 3 primitive results.**

Note: The measurements were performed on the same machine, an 270-2434 600 MHz 2-Way. All results are for a non-secure environment.

## Accelerator for System i

Coinciding with the release of i5/OS V5R4, IBM introduces new entry IBM System i models. The models introduce accelerator technologies and/or L3 cache in order to improve options for clients in the low-end server space. As an overview, the Accelerator for System i affects two 520 Models: (1) 600 CPW with no L3 cache and (2) 1200 CPW with L3 cache. With the Accelerator for System i, the 600 CPW can be accelerated to a 3100 CPW system, whereas the 1200 CPW can be accelerated to 3800 CPW.

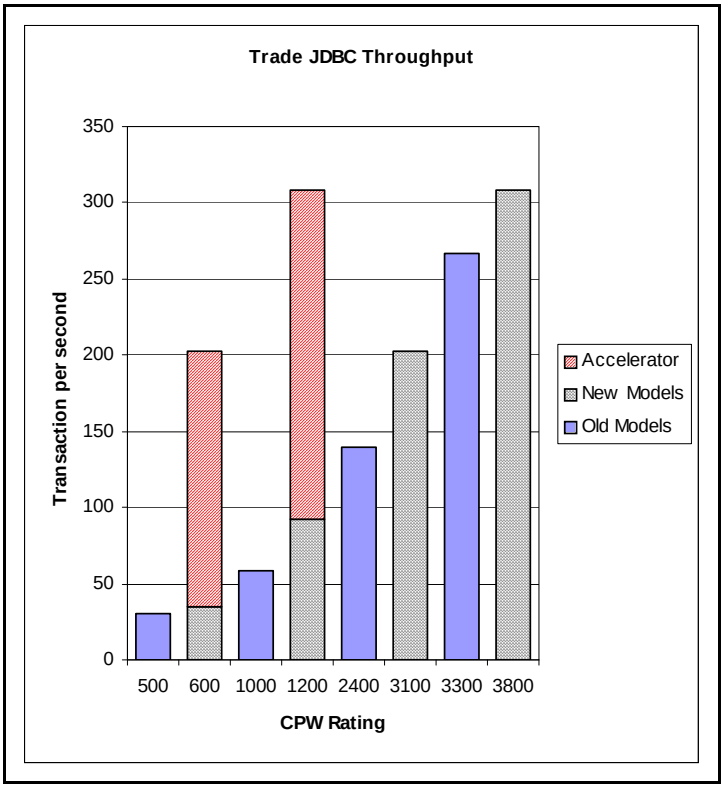



**Figure 6.5** - Accelerator for System i performance data - Capacity comparison (WAS 6.0 running Trade 6).

In order to showcase the abilities of these systems, experiments were completed on WAS 6.0 running Trade 6 to display the benefits. The following information describes the models in the context of both capacity and response time. Results were collected on System i Model 520 with varying feature codes depending on the presence of the Accelerator for System i. With regards to capacity, Figure 6.5 shows the 600 CPW model accelerated to 3100 CPW increases capacity 5.5 times. Additionally, the 1200 CPW model accelerated to 3800 CPW increases capacity 3 times. This provides an extraordinary benefit when running WebSphere Applications.

It is also important to note the benefits of L3 cache. For example, the 1200 CPW model has 2.5 times more capacity than that of the 600 CPW system. Additionally, Java workloads tend to perform better with L3 cache. Thus, besides the benefit of increased capacity, a move from a system with no L3 cache to a system with L3 cache may scale better than CPW ratings would indicate.

Figure 6.6 provides insight into response time information regarding low-end System i models. There are two key concepts that are displayed in the data in Figure 6.6. The first is that Accelerator for System i models can provide substantially better response times than previous models for a single or many users. The 600 CPW accelerated to 3100 CPW reduces the response time by 5 times while the 1200 CPW accelerated to 3800 CPW reduces the response time by 2.5 times. The second idea to note is that the presence of L3 cache has little effect on the response time of a single user. Of course there are benefits of L3 cache, however, the absence of L3 cache does not imply poorer performance with regards to response time.

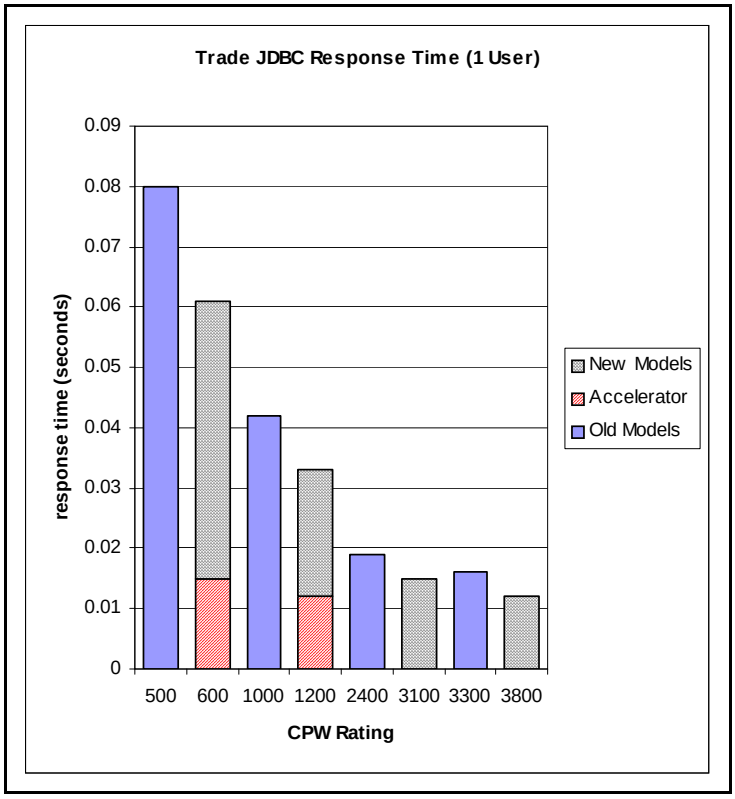



**Figure 6.6** - Accelerator for System i performance data - Single user response time comparison (WAS 6.0 running Trade 6).

**Performance Considerations When Using WebSphere Transaction Processing (XA)**

In a general sense, a transaction is the execution of a set of related operations that must be completed together. This set of operations is referred to as a unit-of-work. A transaction is said to commit when it completes successfully. Otherwise it is said to roll back. When an application needs to access more than one resource (backend) and needs to guarantee job consistency, a global transaction is required to coordinate the interactions among the resources, application and transaction manager, as defined by the XA specification.

WebSphere Application Server is compliant with the XA specification. It uses the two-phase commit protocol to guarantee the All-or-Nothing semantics that either all the resources commit the change permanently or none of them precede the update (rollback). Such a transaction scenario requires the participating resource managers, such as WebSphere JMS/MQ and DB2 UDB, to support the XA specification and provide the XA interface for a two-phase commit. It is the role of the resource managers to manage access to shared resources involved in a transaction and guarantee that the ACID properties (Atomicity, Consistency, Isolation, and Durability) of a transaction are maintained. It is the role of WebSphere Transaction manager to control all of the global transaction logic as defined by the J2EE Standard. Within WebSphere there are two ways of using WebSphere global transaction: Container Managed Transaction (CMT) and Bean Managed Transaction (BMT). With Container Managed, you do not need to write any code to control the transaction behavior. Instead, the J2EE container, WebSphere in this case, controls all the transaction logic. It is a service provided by WebSphere.

When your application involves multiple resources, such as DB2, in a transaction, you need to ensure that you select an XA compliant JDBC provider. For WebSphere on the System i platform you have two options depending on if you are running in a two tier environment (application server and database server on the same system) or in a three tier environment (application server and database server on separate systems). For a two tier environment you would select DB2 UDB for iSeries (Native XA - V5R2 and later). For a three tier environment you would select DB2 UDB for iSeries (Toolbox XA).

However, since the overhead of running XA is quite significant, you should ensure that you do not configure an XA compliant JDBC provider if your application does not require XA functionality. In this case, in a two tier environment you would select DB2 UDB for iSeries (Native - V5R2 and later) and for a three tier environment you would select DB2 UDB for iSeries (Toolbox).

In WebSphere 6.0 the JMS provider was totally rewritten. It is now 100% pure Java and it no longer requires WebSphere MQ to be installed. Also, the datastore for the messaging engine can be configured to store persistent data in DB2 UDB for iSeries. As a result, you can configure your application to share the JDBC connection used by a messaging engine, and the EJB container. This enables you to use one-phase commit (non-XA) transactions since you now have only one resource manager (DB2) involved in a transaction. Previously with 5.1 you had to use XA since the transaction would involve MQ and DB2 resource managers. By utilizing one-phase commit optimization, you can improve the performance of your application.

You can benefit from the one-phase commit optimization in the following circumstances:

- Your application must use the assured persistent reliability attribute for its JMS messages.
- Your application must use CMP entity beans that are bound to the same JDBC data source that the messaging engine uses for its data store.

Restriction: You cannot benefit from the one-phase commit optimization in the following circumstances:
- If your application uses a reliability attribute other than assured persistent for its JMS messages.
- If your application uses Bean Managed Persistence (BMP) entity beans, or JDBC clients.

Before you configure your system, ensure that you consider all of the components of your J2EE application that might be affected by one-phase commits. Also, since the JDBC datasource connection will now be shared by the messaging engine and the EJB container, you need to ensure that you increase the number of connections allocated to the connection pool. To optimize for one-phase commit transactions, refer to the following website:

Http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.pmc.doc/tasks/tjm0280_.html

### WebSphere Application Server V51 Express

For information on WAS V51 Express, please refer to older versions of the Performance Capabilities Reference Manual that can be found here:
http://www.ibm.com/systems/i/solutions/perfmgmt/resource.html

## 6.4 IBM WebFacing

The IBM WebFacing tool converts your 5250 application DDS display files, menu source, and help files into Java Servlets, JSPs, JavaBeans, and JavaScript to allow your application to run in either WebSphere Application Server V5 or V4. This is an easy way to bring your application to either the Internet, or the Intranet, both quickly and inexpensively.

**The Number of Screens processed per second and the number of Input/Output fields per screen** are the main metric to tell how heavy a WebFaced application will be on the WebSphere Application Server. The number of Input/Output fields are simple to count for most of the screens, except when dealing with subfiles. Subfiles can affect the number of input/output fields dramatically. The number of fields in subfiles are significantly impacted by two DDS keywords:

1. SFLPAG - The number of rows shown on a 5250 display.
2. SFLSIZ - The number of rows of data extracted from the database.

When using a DDS subfile, there are 3 typical modes of operation:

1. SFLPAG=SFLSIZ. In this mode, there are no records that are cached. When more records are requested, WebFacing will have to get more rows of data. This is the recommended way to run your WebFacing application.
2. SFLPAG < SFLSIZ. In this mode, WebFacing will get SFLSIZ rows of data at a time. WebFacing will display SFLPAG rows, and cache the rest of the rows. When the user requests more rows with a page-down, WebFacing will not have to access the database again, unless they page below the value of SFLSIZ. When this happens, WebFacing will go back to the database and receive more rows.
3. SFLPAG = (SFLSIZ) * (Number of times requesting the data). This is a special case of option 2 above, and is the recommended approach to run GreenScreen applications. For the first time the page is requested, SFLPAG rows will be returned. If the user performs a page down, then SFLPAG * 2 rows will be returned. This is very efficient in 5250 applications, but less efficient with WebFacing.

Since WebFacing is performance sensitive to the number of input/output fields that are requested from WebFacing, the best option would be the first mode, since this will minimize the number of these fields for each 5250 panel requested through WebFacing. The number of fields for a subfile is the number of rows requested from the database (SFLSIZE) times the number of columns in each row.

Figure 6.7 shows a theoretical algorithm to graphically describe the effect the number of Input/Output fields has on the performance of the WebFaced application. The Y-axis metric is not important, but merely can be used to measure the relative amount of CPU horsepower that the application needs to serve one single 5250 panel. In this case, serving one single panel with 50 I/O fields is approximately one half the CPU horsepower needed to serve one 5250 panel with 350 I/O fields. As you can see, the number of I/O fields dramatically impacts the performance of your WebFacing application, thereby reducing the I/O fields will improve your performance.

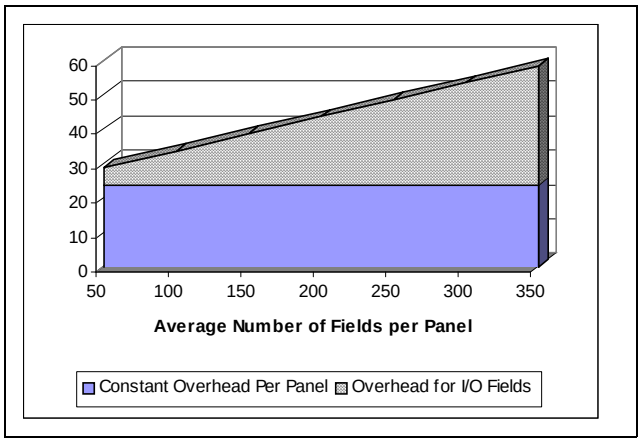



**Figure 6.7** Shows the impact on CPU that the number of I/O fields has per WebFaced panel

**In our studies**, we selected three customer WebFaced applications, one simple, one moderate, and one complex. See table 6.4, for

details on the number of I/O fields for each of these workloads. We ran the workloads on three separate machines (see table 6.5) to validate the performance characteristics with regard to CPW. In our running of the workloads, we tolerated only a 1.5 second **server response time** per panel. This value does not include the time it takes to render the image on the client system, but only the time it took the server to get the information to the client system. The machines that we used are in Table 6.5, and include the 800 and i810 (V5R2 Hardware) and the 170 (V4R4 Hardware). All systems were running OS/400 V5R2.

Some of the results that we saw in our tests are shown in Figure 6.8. This figure shows the scalability across different hardware running the same workload. A user is defined as a client that requests one new 5250 panel every 15 seconds. According to our tests, we see relatively even results across the three machines. The one machine that is a slight difference is the V4R4 hardware (1090 CPW). This slight difference can be explained by release-to-release degradation. Since the CPW measurement were made in V4R4, there have been three major releases, each bringing a slight degradation in performance. This results in a slight difference in CPW value. With this taken into effect, the CPW/User measurement is more in line with the other two machines.

| Name | Average number of I/O Fields / panel |
|---|---|
| Workload A | 37 |
| Workload B | 99 |
| Workload C | 612 |

**Table 6.4** Average number of I/O fields for each workload defined in this section.

Many 5250 applications have been implemented with "best performance" techniques, such as minimized number of fields and amount of data exchanged between the device and the application. Other 5250 applications may not be as efficiently implemented, such as restoring a complete window of data, when it was not required. Therefore it is difficult to give a generalized performance comparison between the same application written to a 5250 device and that application using WebFacing to a browser. In the three workloads that we measured, we saw a significant amount of resource needed to WebFace these applications. The numbers varied from 3x up to 8x the amount of CPU resources needed for the 5250 green screen application.

Use the IBM Systems Workload Estimator to predict the capacity characteristics for IBM WebFacing This site will be updated, more often then this paper, so it will contain the most recent information. The Workload Estimator will ask you to specify a transaction rate (5250 panels per hour) for a peak time of day. It will further attempt to characterize your workload by considering the complexity of the panels and the number of unique panels that are displayed by the JSP. You'll find the tool at: http://www.ibm.com/eserver/iseries/support/estimator. A workload description along with good help text is available on this site. Work with your marketing representative to utilize this tool (also see chapter 23).

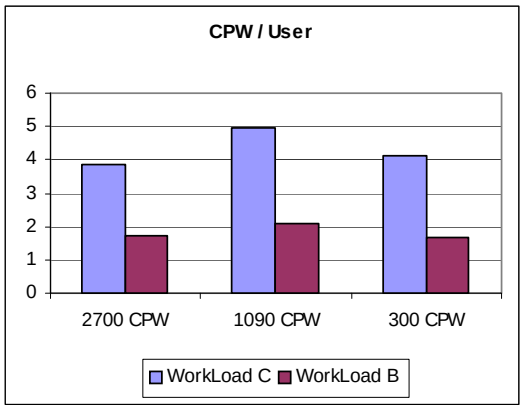



**Figure 6.8** CPW per User across the machines documented in table 6.5

## Version 5.0 of Webfacing

There have been a significant number of enhancements delivered with V5.0 of Webfacing including:

- (Advanced Edition Only) Support for viewing and printing spooled files

- (Advanced Edition Only) Struts-compliant code generated by the WebFacing Tool conversion process which sets the foundation for extending your Webfaced applications using struts-compliant action architecture
- Automatic configuration for UTF-8 support when you deploy to WebSphere Application Server version 5.0
- Support for function keys within window records
- Enhanced hyperlink support
- Improved memory optimization for record I/O processing.
- Support to enable compression to improve response times on slow connections.

The two important enhancements from a performance perspective will be discussed below. For other information related to Webfacing V5.0, please refer to the following website: http://www.ibm.com/software/awdtools/wdt400/about/webfacing.html

**Display File Record I/O Processing**

Display file record I/O processing has been optimized to decrease the WebSphere Application Server runtime memory utilization. This has been accomplished by enhancing the Webfacing runtime to better utilize the java objects required for processing display I/O requests for each end user transaction. Formerly on each record I/O, Webfacing had to create a record data bean object to describe the I/O request, and then create the record bean using this definition to pass the I/O data to the associated JSP. These definition objects were not reused and were created for each user. With the optimization implemented in V5.0, the record bean definitions are now reused and cached so that one instance for each display file record can be shared by all users.

This optimization has decreased the overall memory requirements for Webfacing V5.0 versus V4.0. This memory savings helps reduce the total memory required by the WebSphere Application Server, which is referred to as the JVM Heap Size. The amount of memory savings depends on a number of parameters, such as the complexity of the screens (based on number of fields per screen), the transaction rate, and the number of concurrent end users. On measurements made with approximately 250 users and varying screen complexity, the JVM Heap decreased by approximately 5 % for simple to moderate screens (99 fields per screen) and up to 20 % for applications with more complex screens (600 fields per screen). When looking at the overall memory requirements for an application, the JVM Heap size is just one component. If you are running the back-end application on the same server as the WebSphere Application server, the overall decrease in system memory required for the Webfaced application will be less.

In terms of WebSphere CPU utilization, this optimization offers up to a 10% improvement for complex workloads. However, when taking into account the overall CPU utilization for a Webfaced application (Webfacing plus the application), you can expect equal or slightly better performance with Webfacing V5.0.

**Tuning the Record Definition Cache**

In order to best use the optimization provided by this enhancement, servlet utilities have been included in the Webfacing support to assess cache efficiency, set the cache size, and preload it with the most frequently accessed record definitions. If you do not use the Record Definition Cache, or it is not tuned properly, you will see degraded performance of Webfacing V5.0 versus V4.0.

When set to an appropriate level for the Webfaced application, the Record Definition Cache can provide a decrease in memory usage, and slightly decreased processor usage. The number of record definitions that the cache will retain is set by an initialization parameter in the Webfaced application's deployment descriptor (web.xml). By changing the cache size, the Webfaced application can be tuned for best performance and minimum memory requirements. The cache size determines the number of record data definitions that will be retained in the cache. There is one record data definition for each record format.

| Cache Size | Effect |
|---|---|
| too small | When the cache size is set too small for the Webfaced application it will adversely affect the performance. In this case, the definitions would be cached then discarded before being re-used. There is significant overhead to create the record definitions. |
| correct | With the cache set correctly, 90% of all accessed record data definitions would be retained in the cache with few cache misses for not commonly used records. |
| too large | If the cache is set too large then all record data definitions for the Webfaced application would be cached, likely consuming memory for seldom used definitions. |

In order to determine what is the correct size for a given Webfaced application, the number of commonly used record formats needs to estimated. This can be used as a starting point for setting the cache size. The default size, if no size is specified, would be 600 record data definitions. To set the cache size to something other than the default size, you need to add a session context parameter in the Webfaced application's web.xml file. In the following example the cache size is set to 200 elements, which may be appropriate for a very small application, like the Order Entry example program.

```
<context-param>
  <param-name>WFBeanCacheSize</param-name>
  <param-value>200</param-value>
  <description>WebFacing Record Definition Bean Cache Size</description>
</context-param>
```

**NOTE:** For information on defining a session context parameter in the web.xml file, refer to the WebSphere Application Server Info Center. You can also edit the web.xml file of a deployed application. Typically this file will be located in the following directory for WebSphere V5.0 applications:

/QIBM/UserData/WebAS5/Base/<application-server>/config/cells/..../WEB_INF

And the following directory for WebSphere Express V5.0 applications:

/QIBM/UserData/WebASE/ASE5/<application-server>/config/cells/..../WEB_INF

### Cache Management - Definition Cache Content Viewer

To assist with managing the Record Definition Cache, two servlets can be enabled. One is used to display the elements currently in the cache and the other can be used to load the cache. Both of these servlets are not normally enabled in a WebFacing application in order to prevent mis-use or exposure of data.

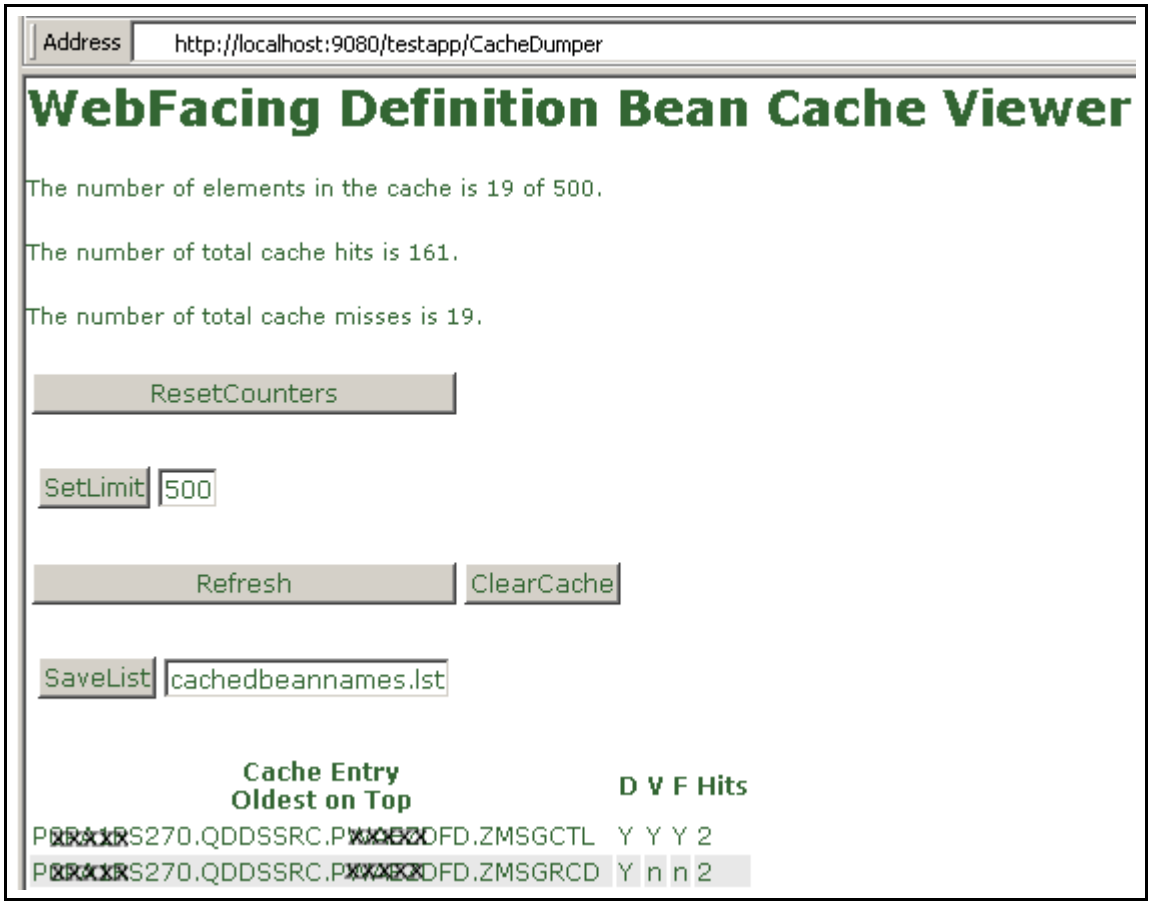
To enable the servlet that will display the contents of the cache, first add the following segments to the Webfaced application's web.xml.

```
<servlet>
  <servlet-name>CacheDumper</servlet-name>
  <display-name>CacheDumper</display-name>
  <servlet-class>com.ibm.etools.iseries.webfacing.diags.CacheDumper</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>CacheDumper</servlet-name>
  <url-pattern>/CacheDumper</url-pattern>
</servlet-mapping>
```

This servlet can then be invoked with a URL like: http://<server>:<port>/<webapp>/CacheDumper.

Then a Web page like that shown below will be displayed. Notice that the total number of cache hits and misses are displayed, as are the hits for each record definition.




Refer to the following table for the functionality provided by the Cache Viewer servlet.

**Cache Viewer Button operations**

| Button | Operation |
|---|---|
| Reset Counters | Resets the cache hit and miss counters back to 0. |
| Set Limit | Temporarily sets the cache limit to a new value. Setting the value lower than the current value will cause the cache to be cleared as well. |
| Refresh | Refresh the display of cache elements. |
| Clear Cache | Drop all the cached definitions. |
| Save List | Save a list of all the cached record data definitions. This list is saved in the RecordJSPs directory of the Webfaced application. The actual record definitions are not saved, just the list of what record definitions are cached. Once the cache is optimally tuned, this list can be used to preload the Record Definition cache. |

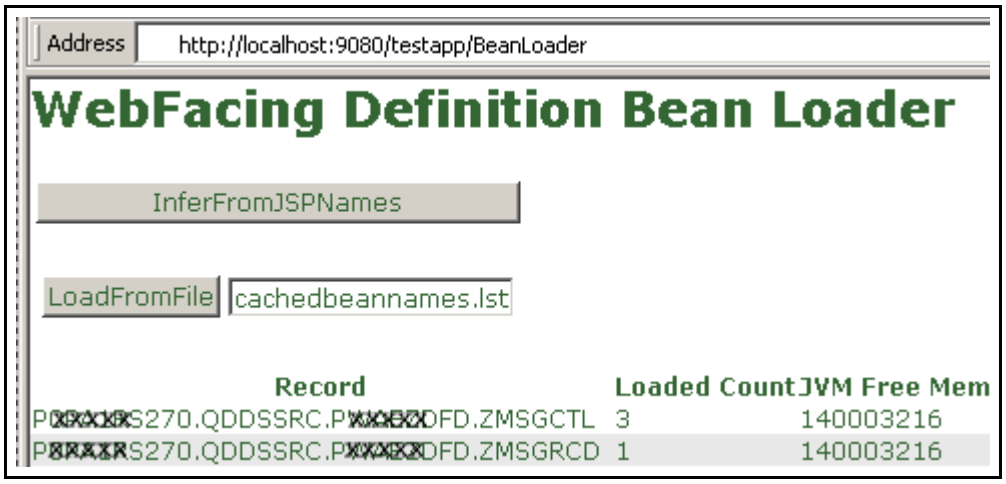
**Cache Management - Record Definition Loader**

As a companion to the Cache Content Viewer tool, there is also a Record Definition Cache Loader tool, which is also referred to as the Bean Loader. This servlet can be used to pre-load the cache to aid in the determination of the optimal cache size, and then finally, to pre-load the cache for production use. To enable this servlet add the following two xml segments in the web.xml file.

```
<servlet>
  <servlet-name>BeanLoader</servlet-name>
  <display-name>BeanLoader</display-name>
  <servlet-class>com.ibm.etools.iseries.webfacing.diags.BeanLoader</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>BeanLoader</servlet-name>
  <url-pattern>/BeanLoader</url-pattern>
</servlet-mapping>
```

Invoking this servlet will present a Web page similar to the following.



Refer to the following table for the functionality provided by the Record Definition Loader servlet.

**Record Definition Loader Button operations**

| Button | Operation |
|---|---|
| Infer from JSP Names | This will cause the loader servlet to infer record definition names from the names or the JSP's contained in the RecordJsps directory. It will not find all the record definitions but it will get most of them. |
| Load from File | This option will load the record definitions listed in a file in the RecordJSPs directory. Typically this file is created with the CacheDumper servlet previously described. |

The Record Definition Loader servlet can also be used to pre-load the bean definitions when the Webfaced application is started. To enable this the servlet definition in the web.xml needs to be updated to define two init parameters: FileName and DisableUI. The FileName parameter indicates the name of the file in the RecordJSPs directory that contains the list of definitions to pre-load the cache with. The DisableUI parameter indicates that the Web UI (as presented above) would be disabled so that the servlet can be used to safely pre-load the definitions without exposing the Webfaced application.

```
<servlet>
 <servlet-name>BeanLoader</servlet-name>
 <display-name>BeanLoader</display-name>
 <servlet-class>com.ibm.etools.iseries.webfacing.diags.BeanLoader</servlet-class>
 <init-param>
  <param-name>FileName</param-name>
  <param-value>cachedbeannames.lst</param-value>
 </init-param>
 <init-param>
  <param-name>DisableUI</param-name>
  <param-value>true</param-value>
 </init-param>
 <load-on-startup>10</load-on-startup>
 </servlet>
```

**Compression**

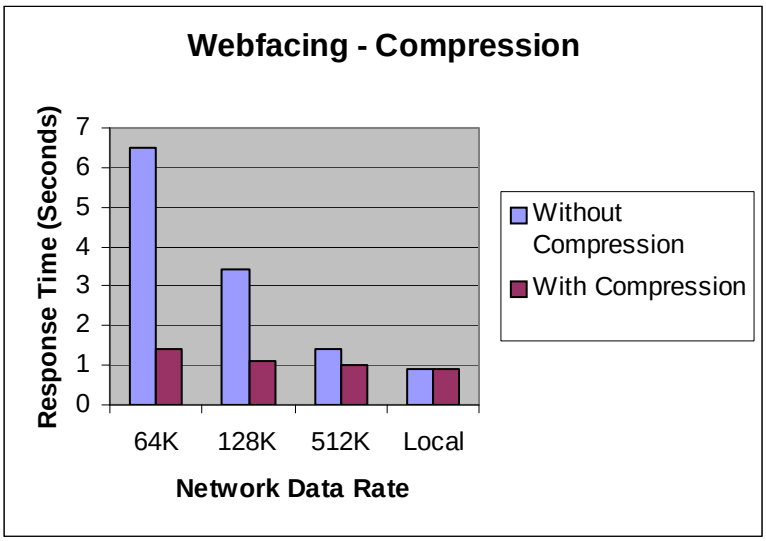
LAN connection speeds and Internet hops can have a large impact on page response times. A fast server but slow LAN connection will yield slow end-user performance and an unhappy customer.

It is very common for a browser page to contain 15-75K of data. Customers who may be running a Webfaced application over a 256K internet connection might find results unacceptable. If every screen averages 60K, the time for that data spent on the wire is significant. Multiply that by several users simultaneously using the application, and page response times will be longer.

There are now two options available to support HTTP compression for Webfaced applications, which will significantly improve response times over a slow internet connection. As of July 1, 2003, compression support was added with the latest set of PTFs for IBM HTTP Server (powered by Apache) for i5/OS (5722-DG1). Also, Version 5.0 of Webfacing was updated to support compression available in

WebSphere Application Server. On System i servers, the recommended WebSphere application configuration is to run Apache as the web server and WebSphere Application Server as the application server. Therefore, it is recommended that you configure HTTP compression support in Apache. However, in certain instances HTTP compression configuration may be necessary using the Webfacing/WebSphere Application Server support. This is discussed below.

The overall performance in both cases is essentially equivalent. Both provide significant improvement for end-user response times on slower Internet connections, but also require additional HTTP/WebSphere Application Server CPU resources. In measurements done with compression, the amount of CPU required by HTTP/WebSphere Application Server increased by approximately 25-30%. When compression is enabled, ensure that there is sufficient CPU to support it. Compression is particularly beneficial when end users are attached via a Wide Area Network (WAN) where the network connection speed is 256K or less. In these cases, the end user will realize significantly improved response times (see chart below). If the end users are attached via a 512K connection, evaluate whether the realized response time improvements offset the increased CPU requirements. Compression should not be used if end users are connected via a local intranet due to the increased CPU requirements and no measurable improvement in response time.




NOTE: The above results were achieved in a controlled environment and may not be repeatable in other environments. Improvements depend on many factors.

**Enabling Compression in IBM HTTP Server (powered by Apache)**

The HTTP compression support was added with the latest set of PTFs for IBM HTTP Server for i5/OS (5722-DG1). For V5R1, the PTFs are SI09287 and SI09223. For V5R2, the PTFs are SI09286 and SI09224.

There is a LoadModule directive that needs to be added to the HTTP config file in order to get compression based on this new support. It looks like this:

LoadModule deflate_module /QSYS.LIB/QHTTPSVR.LIB/QZSRCORE.SRVPGM

You also need to add the directive:

SetOutputFilter DEFLATE

to the container to be compressed, or globally if the compression can always be done. There is documentation on the Apache website on mod_deflate (http://httpd.apache.org/docs-2.0/mod/mod_deflate.html) that has information specific to setting up for compression. That is the best place to look for details. The LoadModule and SetOutputFilter directives are required for mod_deflate to work. Any other directives are used to further define how the compression is done.

Since the compression support in Apache for i5/OS is a recent enhancement , Information Center documentation for the HTTP compression support was not available when this paper was created. The IBM HTTP Server or i5/OS website (http://www.ibm.com/servers/eserver/iseries/software/http/) will be updated with a splash when the InfoCenter documentation has been completed. Until the documentation is available, the information at http://httpd.apache.org/docs-2.0/mod/mod_deflate.html can be used as a reference for tuning how mod_deflate compression is done.

**Enabling Compression using IBM Webfacing Tool and WebSphere Application Server Support**

You would configure compression using the Webfacing/WebSphere support in environments where the internal HTTP server in WebSphere Application Server is used. This may be the case in a test environment, or in environments running WebSphere Express V5.0 on an xSeries Server.

With the IBM WebFacing Tool V5.0, compression is 'turned on' by default. This should be 'turned off' if compression is configured in Apache or if the LAN environment is a local high speed connection. This is particularly important if the CPU utilization of interactive types of users (Priority 20 jobs) is about 70-80% of the interactive capacity. In order to 'turn off' compression, edit the web.xml file for a deployed Web application. There is a filter definition and filter mapping definition that defines compression should be used by the WebFacing application (see below). These statements should be deleted in order to 'turn off' compression. In a future service pack of the WebFacing Tool, it is planned that compression will be configurable from within WebSphere Development Studio Client.

```
<filter id="Filter_1051910189313">
        <filter-name>CompressionFilter</filter-name>
        <display-name>CompressionFilter</display-name>
        <description>WebFacing Compression Filter</description>
        <filter-class>com.ibm.etools.iseries.webfacing.runtime.filters.CompressionFilter</filter-class>
 </filter>
     <filter-mapping id="FilterMapping_1051910189315">
        <filter-name>CompressionFilter</filter-name>
        <url-pattern>/WFScreenBuilder</url-pattern>
 </filter-mapping>
```

**Additional Resources**

The following are additional resources that include performance information for Webfacing including how to setup pretouch support to improve JSP first-touch performance:

**PartnerWorld for Developers Webfacing** website:
http://www.ibm.com/servers/enable/site/ebiz/webfacing/index.html

**IBM WebFacing Tool Performance Update** - This white paper expains how to help optimize WebFaced Applications on IBM System i servers. Requests for the paper require user registration; there are no charges.
http://www-919.ibm.com/servers/eserver/iseries/developer/ebiz/documents/webfacing/

## 6.5 WebSphere Host Access Transformation Services (HATS)

WebSphere Host Access Transformation Services (HATS) gives you all the tools you need to quickly and easily extend your legacy applications to business partners, customers, and employees. HATS makes your 5250 applications available as HTML through the most popular Web browsers, while converting your host screens to a Web look and feel. With HATS it is easy to improve the workflow and navigation of your host applications without any access or modification to source code.

### What's new with V5R4 and HATS 6.0.4

The IBM WebFacing Tool has been delivering reliable and customizable Web-enabled applications for years. Host Access Transformation Services (HATS) has been providing seamless runtime Web-enablement. Now, with the IBM WebFacing Deployment Tool with HATS Technology (WDHT), IBM offers a single product with the power of both technologies.

This offering replaces HATS for iSeries and HATS for System i model 520. For HATS applications created using HATS Toolkit 6.0.4 and deployed to a V5R4 system, you can now connect to the WebFacing Server and eliminate the Online Transaction Processing charge. Without the OLTP requirement for deploying a HATS application to i5/OS starting with V5R4, the overall cost of HATS solutions is significantly reduced. HATS applications can now be deployed to i5/OS Standard Edition.

With WDHT, WebFacing applications can call non-WebFacing applications and those programs will be dynamically transformed for the Web using HATS technology.

### HATS Customization

HATS uses a rules-based engine to dynamically transform 5250 applications to HTML. The process preserves the flow of the application and requires very little technical skill or customization.

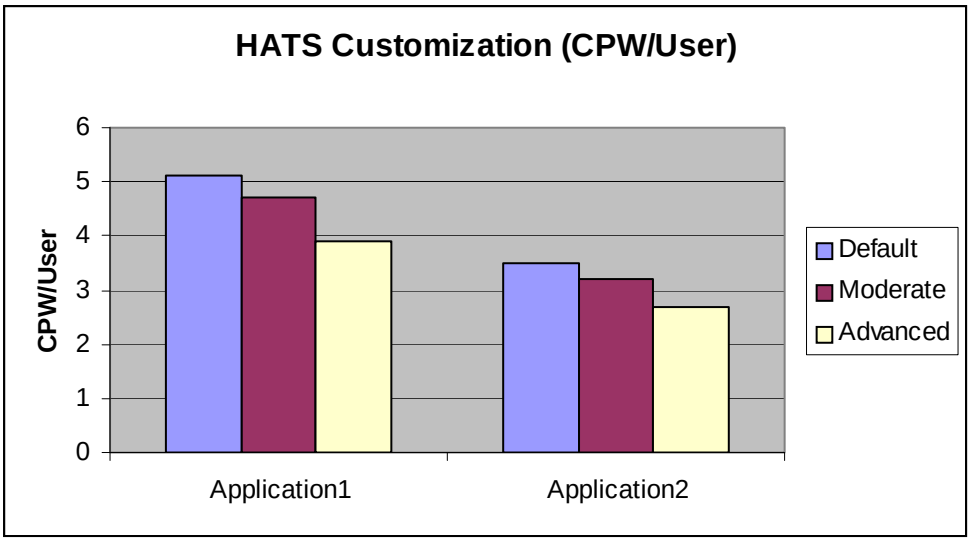
Unless you do explicit customization for an application, the default HATS rules will be used to transform the application interface dynamically at runtime. This is referred to as default rendering. Basically a default template JSP is used for all application screens. There is the capability to change the default template to customize the web appearance, but at runtime the application screens are still dynamically transformed.

As an alternative, you can use HATS studio (built upon the common WebSphere Studio Workbench foundation) to capture and customize select screens or all screens in an application. In this case a JSP is created for each screen that is captured. Then at runtime the first step HATS performs is to check to see if there are any screens that have been captured and identified that match the current host screen. If there are no screen customizations, then the default dynamic transformation is applied. If there is a screen customization that matches the current host screen, then whatever actions have been associated with this screen are executed.

Since default rendering results in dynamic screen transformation at run time, it will require more CPU resources than if the screens of an application have been customized. When an application is customized, JSPs are created so that much of the transformation is static at run time. Based on measurements for a mix of applications using the following levels of customizations, Moderate Customization typically requires 5-10% less CPU as compared to Default Rendering. With Advanced Customization, typically 20-25% less CPU is required as compared to Default Rendering. You have to take into account, though, that

customization requires development effort, while Default Rendering requires minimal development resources.

**Default:** The screens in the application's main path are unchanged.
**Moderate:** An average of 30% of the screens have been customized.
**Advanced**: All screens have been customized.



**IBM Systems Workload Estimator for HATS**

The purpose of the *IBM Systems Workload Estimator (WLE)* is to provide a comprehensive System i sizing tool for new and existing customers interested in deploying new emerging workloads standalone or in combination with their current workloads. The Estimator recommends the model, processor, interactive feature, memory, and disk resources necessary for a mixed set of workloads. WLE was enhanced to support sizing a System i server to meet your HATS workload requirements.

This tool allows you to input an interactive transaction rate and to further characterize your workload. Refer to the following website to access WLE, http://www.ibm.com/estimator/index.html . Work with your marketing representative to utilize this tool, and also refer to chapter 22 for more information.

## 6.6 System Application Server Instance

WebSphere Application Server - Express for iSeries V5 (5722-IWE) is delivered with i5/OS V5R3 providing an out-of-the-box solution for building static and dynamic websites. In addition, V5R3 is shipped with a pre-configured Express V5 application server instance referred to as the System Application Server Instance (SYSINST). The SYSINST has the following IBM supplied system administrative web applications pre-installed[1], providing an easy-to-use web GUI interface to administration tasks:

- iSeries Navigator Tasks for the Web
  Access core systems management tasks and access multiple systems through one System i server from a web browser . Please see the following for more information:
  http://publib.boulder.ibm.com/iseries/v5r3/ic2924/info/rzatg/rzatgoverview.htm
- Tivoli Directory Server Web Administration Tool
  Setup new or manage existing (LDAP) directories for business application data. Please see the following for more information:

http://publib.boulder.ibm.com/iseries/v5r3/ic2924/info/rzahy/rzahywebadmin.htm

The SYSINST is not started by default when V5R3 is installed. Before you begin working with the above functions, the Administration instance of the HTTP Server (port 2001) must be running on your system. The HTTP Admin instance provides an easy-to-use interface to manage web server and application server instances, and allows you to configure the SYSINST to start whenever the HTTP Admin instance is started. The above administrative web applications will then be accessible once the SYSINST is started. Please refer to the following website for more information on how to work with the HTTP Admin instance in configuring the SYSINST:

http://publib.boulder.ibm.com/iseries/v5r3/ic2924/info/rzatg/rzatgprereq.htm

The minimum recommended requirements to support a limited number of users accessing a set of administration functions provided by the SYSINST is 1.25 GB of memory and a system with at least 450 CPW. If you are utilizing only one of the administration functions, such as iSeries Navigator Tasks for the Web or Tivoli Directory Server Web Administration Tool, then the recommended minimum memory is 1 GB. Since the administration functions are integrated with the HTTP Administration Server, the resources for this are included in the minimum recommended requirements. The recommended minimum

[1] Only IBM supplied administrative web applications can be installed in the SYSINST. Customer web applications will need to be deployed to a customer-created application server instance

requirements do not take into account the requirement for other web applications, such as customer applications. You should use IBM Systems Workload Estimator (http://www-912.ibm.com/wle/EstimatorServlet) to determine the system requirements for additional web applications.

## 6.7 WebSphere Portal

The IBM WebSphere Portal suite of products enables companies to build a portal website serving the individual needs of their employees, business partners and customers. Users can sign on to the portal and view personalized web pages that provide access to the information, people and applications they need. This personalized, single point of access to resources reduces information overload, accelerates productivity and increases website usage. As WebSphere Portal supports access through mobile devices, as well as the desktop browser, critical information is always available. Visit the WebSphere Portal InfoCenter for more information:
http://www.ibm.com/developerworks/websphere/zones/portal/proddoc.html

Use the IBM Systems Workload Estimator (Estimator) to predict the capacity characteristics for WebSphere Portal (using the WebSphere Portal workload category). For custom applications, the Workload Estimator will ask you questions about your portal pages served, such as the number of portlets per page and the complexity of each portlet. It will also ask you to specify a transaction rate (visits per hour) for a peak time of day. In addition to custom applications, the Estimator supports Portal Document Manager (PDM) and Web Content Management (WCM) for some releases of WebSphere Portal. Because of potential performance differences between WebSphere Portal releases, the projections for one release cannot be applied to other releases.

- WebSphere Portal Enable 5.1 - Custom applications only.
- WebSphere Portal 6.0 - Custom applications and PDM.
- WebSphere Portal Express 6.0 - Custom applications, PDM, and WCM.

The Estimator is available at: http://www.ibm.com/systems/support/tools/estimator. Extensive descriptions and help text for the Portal workloads are available in the Estimator. Please work with your marketing representative when using the Estimator to size Portal workloads (see also chapter 22).

## 6.8 WebSphere Commerce

Use the IBM Systems Workload Estimator to predict the capacity characteristics for WebSphere Commerce performance (using the Web Commerce workload category). The Workload Estimator will ask you to specify a transaction rate (visits per hour) for a peak time of day. It will further attempt to characterize your workload by considering the complexity of shopping visits (browse/order ratio, number of transactions per user visit, database size, etc.). Recently, the Estimator has also been enhanced to include WebSphere Commerce Pro Entry Edition. The Web Commerce workload also incorporates WebSphere Commerce Payments to process payment transactions. You'll find the tool at:
http://www.ibm.com/eserver/iseries/support/estimator. A workload description along with good help text is available on this site. Work with your marketing representative to utilize this tool (see also chapter 23).

To help you tune your WebSphere Commerce website for better performance on the System i platform, there is a performance tuning guide available at:
http://www-1.ibm.com/support/docview.wss?uid=swg21198883. This guide provides tips and techniques, as well as recommended settings or adjustments, for several key areas of WebSphere and DB2 that are important to ensuring that your website performs at a satisfactory level.

## 6.9 WebSphere Commerce Payments

Use the IBM Systems Workload Estimator to predict the capacities and resource requirements for WebSphere Commerce Payments. The Estimator allows you to predict a standalone WCP environment or a WCP environment associated with the buy visits from a WebSphere Commerce estimation. Work with your marketing representative to utilize this tool. You'll find the tool at: http://www.ibm.com/eserver/iseries/support/estimator.

**Workload Description**: The PayGen workload was measured using clients that emulate the payment transaction initiated when Internet users purchase a product from an e-commerce shopping site. The payment transaction includes the Accept and Approve processing for the initiated payment request. WebSphere Commerce Payments has the flexibility and capability to integrate different types of payment cassettes due to the independent architecture. Payment cassettes are the plugins used to accommodate payment requirements on the Internet for merchants who need to accept multiple payment methods. For more information about the various cassettes, follow the link below: http://www-4.ibm.com/software/webservers/commerce/paymentmanager/lib.html

**Performance Tips and Techniques**:

1. **DTD Path Considerations:** When using the Java Client API Library (CAL), the performance of the WebSphere Commerce Payments can be significantly improved if the merchant application specifies the dtdPath parameter when creating a PaymentServerClient. When this parameter is specified, the overhead of sending the entire IBMPaymentServer.dtd file with each response is avoided. The dtdPath parameter should contain the path of the locally stored copy of the IBMPaymentServer.dtd file. For the exact location of this file, refer to the *Programmer's Guide and Reference* at the following link: http://www-4.ibm.com/software/webservers/commerce/payment/docs/paymgrprog22as.html

2. **Other Tuning Tips:** More performance tuning tips can be found in the *Administrator's Guide* under Appendix D at the following link: http://www-4.ibm.com/software/webservers/commerce/payment/docs/paymgradmin22as.html

3. **WebSphere Tuning Tips:** Please refer to the WebSphere section in section 6.2, for a discussion on WebSphere Application Server performance as well as related web links**.**

## 6.10 Connect for iSeries

IBM Connect for iSeries is a software solution designed to provide System i server customers and business partners a way to communicate with an eMarketplace. Connect for iSeries was developed as a software integration framework that allows customers to integrate new and existing back-end business applications with those of their trading partners. It is built on industry standards such as Java, XML and MQ Series.

The framework supports plugins for multiple trading partner protocols. Connect for iSeries also provides pluggable connectors that make it easy to communicate to various back-end applications through a variety

of access mechanisms. Please see the Connect for iSeries white paper located at the following URL for more information on Connect for iSeries.
http://www-1.ibm.com/servers/eserver/iseries/btob/connect/pdf/whtpaperv11.pdf

**"B2B New Order Request" Workload Description:** This workload is driven by a program that runs on a client work station that simulates multiple Web users. These simulated users send in cXML "New Order Request" transactions to the System i server by issuing an HTTP post which includes the cXML New Order Request file as the body of the message. Besides the Connect for iSeries product, other files and back-end application code exist to complete this transaction flow. For this workload, XML validation was disabled for both requests and response flows. The intention of this workload is to drive the server with a heavy load and to quantify the performance of Connect for iSeries.

**Measurement Results:** One of the main focal points was to evaluate and compare the differences between the back-end application connector types. The five connector types compared were the Java, JDBC, MQ Series, Data Queue, and PCML connectors. The graphs below illustrates the relative capacities for each of the connector types. Please visit this link to learn about differences in connector types.
http://www-1.ibm.com/servers/eserver/iseries/btob/connect/pdf/whtpaperv11.pdf

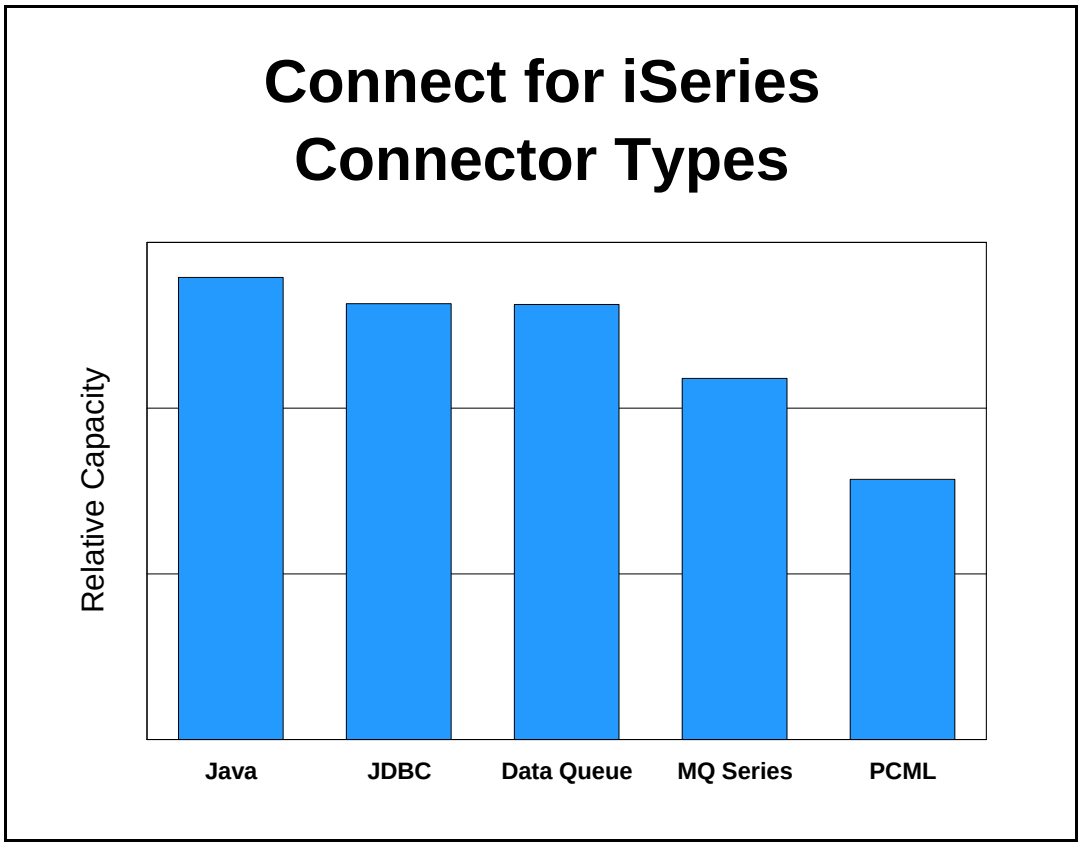



*Figure 6.9 Connect for iSeries - Connector Types*

**Performance Observations/Tips:**

1. **Connector relative capacity:** The different back-end connector types are meant to allow users a simple way to connect the Connect for iSeries product to their back-end application. Your choice in a connector type may be dictated by several factors. Clearly, one of these factors relate to your existing back-end application and the programming language it is written in. This, in itself, may limit your choice for a back-end connector type. Please see the Connect for iSeries white paper to assist you in understanding the different connector types.
   http://www-1.ibm.com/servers/eserver/iseries/btob/connect/pdf/whtpaperv11.pdf

   Performance was measured for a simple cXML New Order Request. The Java connector performance may vary depending on the code you write for it. All connectors "mapped" approximately the same number of "fields" to make a fair comparison. The PCML connector has overhead associated with it in starting a job for each transaction via "SBMJOB". You can pre-start a pool of these jobs which may increase performance for this connector type.

2. **XML Validation:** XML validation should be avoided when not needed. Although many businesses will decide to have this feature on (you may not be able to assume the request is both "well formed and validated") there are significant performance implications with this property "on". One thought would be to enable XML validation during your testing phase. Once your confident that your trading partner is sending valid and well-formed XML, you may want to disable XML validation to improve performance.

3. **Tracing:** Try to avoid tracing when possible. If enabled, it will impact your performance. However, in some cases it is unavoidable (e.g. trouble shooting problems).

4. **Management Central Logging:** This feature will log transaction data to be queried and viewed with Management Central. Performance is impacted with this feature "on" and must be taken into consideration when deciding to use this feature.

5. **MQ Series Management Central Audit Queue:** Due to the fact that the Management Central Auditing logs messages into a MQ Series queue for processing, the default queue size may not be large enough if you run at a very high transaction rate. This can be adjusted by issuing wrkmqm and selecting the queue manager for your Connect for iSeries instance, selecting option 18 (work with queues) on that queue manager, selecting option 2 (change) and increasing the Maximum Queue Depth property. This property, when enabled, added approximately 15% overhead to the "B2B New Order Request" workload.

6. **Recovery (Check pointing):** Enabling transaction recovery adds significant overhead. This should be avoided when not needed. This property when enabled added approximately 50% overhead to the "B2B New Order Request" workload.

7. **MQ Series Connector Queue Configuration:** By default, in MQ Series 5.2, the queue manager uses a single threaded listener which submits a job to handle each incoming connection request. This has performance implications also. The queue manager can be changed to having a multithreaded listener by adding the following property to the file
   \QIBM\UserData\mqm\qmgrs\QMANAGERNAME\qm.ini
   Channels:
   ThreadedListener=Yes
   The multithreaded listener can boast a higher throughput, but the single threaded listener is able to handle many more concurrent connections. Please see MQ Series site for help with MQ Series.
   http://www-4.ibm.com/software/ts/mqseries/messaging/

# Chapter 7. Java Performance

Highlights:

- Introduction
- What's new in V6R1
- IBM Technology for Java (32-bit and 64-bit)
- Classic VM (64-bit)
- Determining Which JVM to Use
- Capacity Planning
- Tips and Techniques
- Resources

## 7.1 Introduction

Beginning in V5R4, IBM began a transition to a new VM implementation for i5/OS, IBM Technology for Java, to replace the Classic VM. This transition continues in V6R1 with the introduction of a 64-bit version of IBM Technology for Java, providing a new solution for Java applications which require large amounts of memory. The transition is expected to be completed in the next version of i5/OS, which will no longer support the Classic VM. In the mean time, one of the key performance -related decisions for i5/OS Java users is which JVM to use.

Earlier versions of this document have followed the performance of Java from its infancy to maturity. Early Java applications were often a departure from the traditional OS/400 application architecture, with custom application code responsible for a large portion of the CPU used by the application. Therefore, earlier versions of this document emphasized micro-optimizations – relatively small (though often pervasive) changes to application code to improve performance.

Today's Java applications, however, typically rely on a variety of system services such as JDBC, encryption, and security provided by i5/OS, the Java Virtual Machine (VM), and WebSphere Application Server (WAS), along with other products built on top of WebSphere. As a result, many Java applications now spend far more time in these system services than in custom code. For many applications, this means that performance depends mainly on the performance of IBM code (i5/OS, the Java VM, WebSphere, etc.) and the way that these services are used by the application. Micro-optimizations can still be important in some cases, but are not as critical as they have been in the past.

Tuning is also important for getting good performance in the Java environment. Tuning garbage collection is perhaps the most common example. Thread and connection pool tuning is also frequently important. Proper tuning of i5/OS can also make a big impact on Java application performance.

## 7.2 What's new in V6R1

In V5R4 IBM introduced IBM Technology for Java, a new VM implementation built on technology used across all of the IBM Systems platforms. In V5R4 only a 32-bit version of IBM Technology for Java was supported; in V6R1, a new 64-bit version of IBM Technology for Java is also available, providing a new

option for Java applications which require large amounts of memory. The Classic VM remains available in V6R1, but future i5/OS releases are expected to support only IBM Technology for Java.

The default VM in V6R1 is IBM Technology for Java 5.0, 32-bit. Other supported versions of IBM Technology for Java include 5.0 64-bit, 6.0 32-bit, and 6.0 64-bit. (6.0 versions will require the latest PTFs to be loaded.) The Classic VM supports Java versions 1.4, 5.0, and 6.0. In V5R4, the default VM is Classic 1.4. Classic 1.3, 5.0, and 6.0 are also supported, as well as IBM Technology for Java 5.0 32-bit and 6.0 32-bit.

Java applications using the Classic VM will generally have equivalent performance between V5R4 and V6R1, although applications which use JDBC to access database may see some improvement. The Classic VM no longer supports Direct Execution (DE) in V6R1; all applications will run with the Just In Time (JIT) compiler. As a result, applications which previously used DE may see some performance difference (usually a significant improvement) when moving to V6R1. Because the same underlying VM is used for all versions of Classic, most applications will see little performance difference between the different JDK levels.

V6R1 offers significant performance improvements over V5R4 when running IBM Technology for Java -- on the order of 10% for many applications, with larger improvements possible when using the -Xlp64k flag to enable 64k pages. In addition, there are substantial performance improvements when moving from IBM Technology for Java 5.0 to 6.0. Performance improvements are frequently introduced in PTFs.

Recent generations of hardware have greatly improved the performance of computationally-intensive applications, which include most Java applications. Since their introduction in V5R3, System i5 servers employing POWER5 processors – models 520, 550, 570, and 595 – have a proven record of providing excellent performance for Java applications. The POWER5+ models introduced with V5R4 build on this success, with performance improvements of up to 30% for the same number of processors in some models. The new POWER6 models introduced in 2007 provide further performance gains, especially for Java applications, which tend to be computationally intensive.

The 515 and 525 models introduced in April, 2007 all include a minimum of 3800 CPW and include L3 cache. These systems deliver solid Java performance at the low-end. Other attractive options at the low-end are the 600 and 1200 CPW models (520-7350 and 520-7352), which have an accelerator feature which allow them to be upgraded to 3100 and 3800 CPW (non-interactive), respectively.

## 7.3 IBM Technology for Java (32-bit and 64-bit)

IBM's extensive research and development in Java technology has resulted in significant advances in performance and reliability in IBM's Java implementations. Many of these advances have been incorporated into the i5/OS Classic VM, but in order to make the latest developments available to System i customers as quickly as possible, IBM introduced a new 32-bit implementation of Java to i5/OS in V5R4. This VM is built on the same technology as IBM's VMs for other platforms, and provides a modular and flexible base for further improvements in the future. In V6R1, a 64-bit version of the same VM is also available.

IBM Technology for Java currently supports Java 5.0 (JDK version 1.5) and (with the latest PTFs) Java 6 (JDK version 1.6). Older versions of the JDK are only supported with the Classic 64-bit VM.

On i5/OS, IBM Technology for Java runs in i5/OS Portable Application Solutions Environment (i5/OS PASE) with either a 32-bit (for the 32-bit VM) or 64-bit (for the 64-bit VM) environment. Due to sophisticated memory management, both the 32-bit and 64-bit VMs provide a significant reduction in memory requirements over the Classic VM for most applications. Because the 32-bit VM uses only 4 bytes (instead of 8 bytes) for object references, applications will have an even smaller memory footprint with the 32-bit VM; however, the 32-bit address space leads to a maximum heap size of 2.5 - 3.25 GB, which may not be enough memory for some applications.

Because IBM Technology for Java shares a common implementation with IBM's VMs on other platforms, the available tuning parameters are essentially the same on i5/OS as on other platforms. This will require some adjustment for users of the i5/OS Classic VM, but may be a welcome change for those who work with Java on multiple platforms.

Some of the key areas to be aware of when considering use of IBM Technology for Java are described below.

### Native Code

Because IBM Technology for Java runs in i5/OS PASE, there is some additional overhead in calls to native ILE code. This may affect performance of certain applications which make calls to native ILE code through the Java Native Interface (JNI). Calls to certain operating system services, such as IFS file access and socket communication, may also have some additional overhead, although the overhead should be minimal for applications with a typical use of these services. Conversely, JNI calls to PASE native methods will have less overhead than they did with the Classic VM, offering a performance improvement for some applications.

The performance impact for JNI method calls to ILE will depend on the frequency of JNI calls and the complexity of the native methods. If the calls are infrequent, or if the native methods are very complex (and therefore take a long time to execute), the increased overhead may not make a big difference in the overall performance of the application. Applications which make frequent calls to simple native methods may see a more significant performance impact compared to the 64-bit Classic VM.

For some applications, it may be possible to port these native methods to run in i5/OS PASE rather than in ILE, greatly reducing the overhead of the native call. In other cases, it may be possible to modify the application to require fewer JNI calls.

### Garbage Collection

Recommendations for Garbage Collector (GC) tuning with the i5/OS Classic VM have always been a bit different from tuning recommendations for Java VMs on other platforms. While the main GC tuning parameters (initial and max heap size) have the same names as the key parameters for other VMs, and are set in the same way when running Java from qsh (-Xms and -Xmx), the meaning of these parameters in the Classic 64-bit VM is significantly different. However, with IBM Technology for Java these parameters mean the same thing that they do in IBM VMs on other platforms. Many users will welcome this commonality; however, it does make the transition to the new VM a bit more complicated. The move from a 64-bit VM to a 32-bit VM also complicates matters somewhat, as the ideal heap size will be significantly lower in a 32-bit VM than in a 64-bit VM.

Fortunately, it is not too difficult to come up with parameter values which will provide good performance. If you are moving an application from the Classic VM to IBM Technology for Java, you can use a tool like DMPJVM or verbose GC to determine how large the heap grows when running your application. This value can be used as the maximum heap size for 64-bit IBM Technology for Java; in 32-bit IBM Technology for Java, about 75% of this value is a reasonable starting point. For example, if your application's heap grows to 256 MB when running in the Classic VM, try setting the maximum heap size to 192 MB when running in the 32-bit VM. The initial heap size can be set to about half of this value – 96 MB in our example. These settings are unlikely to provide the best possible performance or the smallest memory footprint, but the application should run reasonably well. Additional performance tests and tuning could result in better settings.

If your application also runs on IBM VMs on other platforms, such as AIX, then you might consider trying the GC parameters from those platforms as a starting point when using IBM Technology for Java on i5/OS.

If you are testing a new application, or aren't certain about the performance characteristics of an existing application running in the Classic 64-bit VM, start by running the application with the default heap size parameters (currently an initial heap size of 4 MB and a maximum of 2 GB). Run the application and see how large the heap grows under a typical peak load. The maximum heap size can be set to this value (or perhaps slightly larger). Then the initial heap size can be increased to improve performance. The optimal value will depend on the application and several other factors, but setting the initial heap size to about 25% of the maximum heap size often provides reasonable performance.

Keep in mind that the maximum heap size for the 32-bit VM is 3328 MB. Attempting to use a larger value for the initial or maximum heap size will result in an error. The maximum heap size is reduced when using IBM Technology for Java's "Shared Classes" feature or when files are mapped into memory (via the java.nio APIs). The maximum heap size can also be impacted when running large numbers of threads, or by the use of native code running in i5/OS PASE, since the memory used by this native code must share the same 32-bit address space as the Java VM. As a result, many applications will have a practical limit of 3 GB (3072 MB) or even less. Applications with larger heap requirements may need to use one of the 64-bit VMs (either IBM Technology for Java or the Classic VM).

When heap requirements are not a factor, the 64-bit version of IBM Technology for Java will tend to be slightly slower (on the order of 10%) than 32-bit with a somewhat larger (on the order of 70%) memory footprint. Thus, the 32-bit VM should be preferred for applications where the maximum heap size limitation is not an issue.

## 7.4 Classic VM (64-bit)

The 64-bit Classic Java Virtual Machine continues to be supported in V6R1, though most applications should begin migrating to IBM Technology for Java to take advantage of its performance benefits. The integration of the Classic VM into i5/OS provides some unique features and benefits, although this can result in some confusion to users who are familiar with running Java applications on other platforms. Some of the performance-related features you may need to be aware of are described below.

### JIT Compiler

Interpreting the platform-neutral bytecodes of a Java class file, bytecode by bytecode, is one valid and robust way to execute Java object code; it is not, however, the fastest way. To approach optimal Java

performance, it pays to apply analysis and optimizations to the Java bytecodes, and the resulting machine code.

One approach to optimizing Java bytecode involves analyzing the object code “ahead of time” – before it is actually running. This “ahead-of-time” (AOT) compiler technology was used exclusively by the original AS/400 Java Virtual Machine, whose success proved the power of such an approach.

However, any static AOT analysis suffers one fatal flaw: in a dynamically loading language such as Java, it is impossible for an AOT compiler to know exactly what the environment will look like when the code is actually being executed. Certain valuable optimizations – such as inter-class method inlining or parameter-passing optimizations – cannot be made without adding extra checks to ensure that the optimization is still valid at run-time. While these checks are trimmed down as much as possible, some amount of overhead is unavoidable.

When Java was first introduced to the AS/400 it used an AOT compilation approach, with a combination of bytecode interpretation and Direct Execution (DE) programs to statically optimize Java code for the OS/400 environment, with startup and runtime performance usually significantly faster than what other Java implementations at the time could provide.

Later, “Just-In-Time” (JIT) compiler technology was introduced in many Java VMs. Unlike AOT compilation, JIT compiles Java bytecodes to machine code on-the-fly as the application is running. While this introduces some overhead as the compilation occurs, the compiler can optimize much more aggressively, because it knows the exact state of the system it is compiling for.

Over time, JIT compilation technology improved and was implemented alongside DE in the i5/OS Classic VM. JIT performance overtook DE in the V5R2 time frame for most applications, and has continued to improve at a faster rate. In V6R1, support for DE was eliminated, so the JIT will be used for all Java applications.

Despite the improvements to JIT for both runtime and startup performance, startup time does tend to be slightly longer for JIT than DE. Beginning in V5R2, the Mixed Mode Interpreter (MMI) is used to interpret code until it has been executed a number of times (2000 by default, can be overridden by setting the system property `os400.jit.mmi.threshold`) before JIT compiling it, resulting in improved startup time. V5R3 introduced asynchronous JIT compilation, which further improved startup time, especially on multiprocessor systems. As a result of these and other improvements, many applications will no longer see a significant difference in startup time between DE and JIT. Even if startup time is a bit longer with JIT, the improvement in runtime performance may be worth it, especially for long-running applications which don’t start up frequently.

Prior to V6R1, the default execution mode is “`jitc_de`”, which uses DE for Java classes which already have DE programs, and JIT for classes which do not. Notably, JDK classes are shipped with DE program objects created, and will therefore use DE by default. Set the system property `java.compiler` to `jitc` to force JIT to be used for all Java code in your application. (See InfoCenter for instructions about setting Java system properties.)

Note that even when running with the JIT, the VM will have to create a Java program object (with optimization level *INTERPRET) the first time a particular Java class is used on the system, if one does not already exist. Creation of this program object is much faster than creating a full DE program, but it may still make a noticeable difference in startup time the first time your application is used, particularly in

applications with a large number of classes. Running CRTJVAPGM with OPTIMIZE(*INTERPRET) will create this program ahead of time, making the first startup faster.

### Garbage Collection

Java uses Garbage Collection (GC) to automatically manage memory by cleaning up objects and memory when they are no longer in use. This eliminates certain types of memory leaks which can be caused by application bugs for applications written in other languages. But GC does have some overhead as it determines which objects can be collected. Tuning the garbage collector is often the simplest way to improve performance for Java applications.

The Garbage Collector in the i5/OS Classic VM works differently from collectors in Java VMs on other platforms, and therefore must be tuned differently. There are two parameters that can be used to tune GC: GCHINL (-Xms) and GCHMAX (-Xmx). The JAVA/RUNJVA commands also include GCHPTY and GCHFRQ, but these parameters are ignored and have no effect on performance.

The Garbage Collector runs asynchronously in one or more background threads. When a GC cycle is triggered, the Garbage Collector will scan the entire Java heap, and mark each of the objects which can still be accessed by the application. At the end of this "mark" phase, any objects which have not been marked are no longer accessible by the application, and can be deleted. The Garbage Collector then "sweeps" the heap, freeing the memory used by all of these inaccessible objects.

A GC cycle can be triggered in a few different ways. The three most common are:
1. An amount of memory exceeding the collection threshold value (GCHINL) has been allocated since the previous GC cycle began.
2. The heap size has reached the maximum heap value (GCHMAX).
3. The application called *java.lang.System.gc( )* [not recommended for most applications]

The collection threshold value (GCHINL or -Xms, often referred to as the "initial heap size") is the most important value to tune. The default size for V5R3 and later is 16 MB. Using larger values for this parameter will allow the heap to grow larger, which means that GC will run less frequently, but each cycle will take longer. Using smaller values will keep the heap smaller, but GC will run more often. The best value depends on the number, size, and lifetime of objects in your application as well as the amount of memory available to the application. Most applications will benefit from using a larger collection threshold value – 96 MB is reasonable for many applications. For WebSphere applications on larger systems, heap threshold values of 512 MB or more are not uncommon.

The maximum heap size (GCHMAX, or -Xmx) specifies the largest that the heap is allowed to grow. If the heap reaches this size, a synchronous garbage collection will be performed. All other application threads will have to wait while this GC cycle occurs, resulting in longer response times. If this synchronous GC cycle is not able to free up enough memory for the application to continue, an OutOfMemoryError will be thrown. The default value for this parameter is *NOMAX, meaning that there is no limit to the heap size. In practice, a well behaved application will settle out to some steady state heap size, so *NOMAX does not mean that the heap will grow infinitely large. Most applications can leave this parameter at its default value.

One important consideration is to not allow the Java heap to grow beyond the amount of physical memory available to the application. For example, if the application is running in the *BASE memory pool with a size of 1 GB, and the heap grows to 1.5 GB, the paging rate will tend to get quite high, especially when a GC cycle is running. This will show up as non-database page faults on the WRKSYSSTS command

display; rates of 20 to 30 faults per second are usually acceptable, but larger values may indicate a performance problem. In this case, the size of the memory pool should be increased, or the collection threshold value (GCHINL or -Xms) should be decreased so the heap isn't allowed to grow as large. In many cases the scenario may be complicated by the fact that multiple applications may be running in the same memory pool. Therefore, the total memory requirements of all of these applications must be considered when setting the pool size. In some environments it may be useful to run key Java applications in a private pool in order to have more control over the memory available to these applications.

In some cases it may also be helpful to set the maximum heap size to be slightly larger than the memory pool size. This will act as a safety net so that if the heap does grow beyond the memory pool size, it will not cause high paging rates. In this case, the application will probably not be usable (due to the synchronous garbage collection cycles and OutOfMemoryErrors that may occur), but it will have less impact on any other applications running on the system.

A final consideration is the application's use of objects. While the garbage collector will prevent certain types of memory leaks, it is still possible for an application to have an "object leak". One common example is when the application adds new objects to a List or Map, but never removes the objects. Therefore the List or Map continues to grow, and the heap size grows along with it. As this growth continues, the garbage collector will begin taking longer to run each cycle, and eventually you may exhaust the physical memory available to the application. In this case, the application should be modified to remove the objects from the List or Map when they are no longer needed so the heap can remain at a reasonable size. A similar example involves the use of caches inside the application. If these caches are allowed to grow too large, they may consume more memory than is physically available on the system. Using smaller cache sizes may improve the performance of your application.

### Bytecode Verification

In order to maintain system stability and security, it is important that Java bytecodes are verified before they are executed, to ensure that the bytecodes don't try to do anything not allowed by the Java VM specification. This verification is important for any Java implementation, but especially critical for server environments, and perhaps even more so on i5/OS where the JVM is integrated into the operating system. Therefore, in i5/OS, bytecode verification is not only turned on by default, but it is impossible to turn it off. While the bytecode verification step isn't especially slow, it can impact startup time in certain cases – especially when compared to VMs on other platforms which may not do bytecode verification by default. In most cases, full bytecode verification can be done just once for a class, and the resulting JVAPGM objects saved with its corresponding class or jar file as long as the class doesn't change.

However, when user classloaders are used to load classes, the VM may not be able to locate the file from which the class was loaded (in particular, if the standard URLClassLoader mechanism is not being used by the user classloader). In this case, the bytecode verification cache is used to minimize the cost of bytecode verification.

In V5R3 and later the bytecode verification cache is enabled by default, and tuning is usually unnecessary. In V5R2 and earlier releases the cache was disabled by default, and tuning was sometimes necessary. The cache can be turned on by specifying a valid value (e.g., /QIBM/ProdData/Java400/QDefineClassCache.jar) for the os400.`define.class.cache.file` system property. It may also be helpful to set `os400.define.class.cache.maxpgms` to a value of around 20000, since the default of 5000 had been shown to be too small for many applications. In V5R3 and

later releases the cache is enabled and the maxpgms set to 20000 by default, so no adjustment is usually necessary.

The verification cache operates by caching JVAPGMs that have been dynamically created for dynamically loaded classes. When the verification cache is not operating, these JVAPGMs are created as temporary objects, and are deleted as the JVM shuts down. When the verification cache is enabled, however, these JVAPGMs are created as persistent objects, and are cached in the (user specified) machine-wide cache file. If the same (byte-for-byte identical) class is dynamically loaded a second time (even after the machine is re-IPLed), the cached JVAPGM for that class is located in the cache and reused, eliminating the need to verify the class and create a new JVAPGM (and eliminating the time and performance impact that would be required for these actions). Older JVAPGMs are "aged out" of the cache if they are not used within a given period of time (default is one week).

In general, the only cost of enabling the verification cache is a modest amount of disk space. If it turns out that your application is not using one of the problem user class loaders, the cache will have no impact, positive or negative, while if your application is using such a class loader then the time taken to create and cache the persistent JVAPGM is only slightly more than the time required to create a temporary JVAPGM. With next to zero downside risk, and a decent potential to improve performance, the verification cache is well worth a try.

Maintenance is not a problem either: if the source for a cached JVAPGM is changed, the currently-cached version will simply "age out" (since its class will no longer be a byte-for-byte match), and a new JVAPGM will be silently created and cached. Likewise, the cache doesn't care about JDK versions, PTFs installed, application upgrades, etc.

---

## 7.5 Determining Which JVM to Use

Beginning in V5R4, applications can run in either the Classic 64-bit VM or with IBM Technology for Java (32-bit only in V5R4, 32-bit or 64-bit in V6R1). Both VM implementations provide a fully compliant implementation of the Java specifications, and pure Java applications should be able to run without changes in either VM by setting the JAVA_HOME environment variable appropriately. (See InfoCenter for details on specifying which VM will be used to execute a Java program.) However, some applications may have dependencies which will prevent them from working on one of the VM implementations.

In general, applications should use 32-bit IBM Technology for Java when possible. Applications which require larger heaps than can be managed with a 32-bit VM should use 64-bit IBM Technology for Java (on V6R1). The Classic VM also remains available for cases where IBM Technology for Java is not appropriate and to ease migration from older releases.

Some factors to consider include:

### Functional Considerations

1. IBM Technology for Java was introduced in i5/OS V5R4M0. Older versions of OS/400 and i5/OS support only the Classic VM.
2. IBM Technology for Java only supports Java 5.0 (JDK 1.5) and higher. Older versions of Java (1.4, 1.3, etc.) are not supported. While the Java versions are generally backward compatible, some

libraries and environments may require a particular version. The Classic VM continues to support JDK 1.3, 1.4, 1.5 (5.0), and 1.6 (6.0) in V5R4, and JDK 1.4, 1.5 (5.0), and 1.6 (6.0) in V6R1.
3. The Classic VM supported an i5/OS-specific feature called Adopted Authority. IBM Technology for Java does not support this feature, so applications which require Adopted Authority must run in the Classic VM. This will not affect most applications. Applications which do use Adopted Authority should consider migrating to APIs in IBM Toolbox for Java which can serve a similar purpose.
4. Java applications can call native methods through the Java Native Interface (JNI) with either VM. When using IBM Technology for Java, these native programs must be compiled with teraspace storage enabled. In addition, whenever a buffer is passed to JNI functions such as *GetxxxArrayRegion*, the pointer must point to teraspace storage.
5. When using 32-bit IBM Technology for Java runs in a 32-bit PASE environment, any PASE native methods must also be 32-bit. With 64-bit IBM Technology for Java, PASE native methods must be 64-bit. The Classic VM can call both 32-bit and 64-bit PASE native methods. All of the VMs can call ILE native methods as well.

**Performance Considerations**

1. When properly tuned, applications will tend to use significantly less memory when running in IBM Technology for Java than in the Classic VM. Performance tests have shown a reduction of 40% or more in the Java heap for most applications when using the 32-bit IBM Technology for Java VM, primarily because object references are stored with only 4 bytes (32 bits) rather than 8 bytes (64 bits). Therefore, an application using 512 MB of heap space in the 64-bit Classic VM might require 300 MB or even less when running in 32-bit IBM Technology for Java. The difference between the Classic VM and 64-bit IBM Technology for Java is somewhat less noticable, but 64-bit IBM Technology for Java will still tend to have a smaller footprint than Classic for most applications.
2. The downside to using a 32-bit address space is that it limits the amount of memory available to the application. As discussed above, the 32-bit VM has a maximum heap size of 3328 MB, although most applications will have a practical limit of 3 GB or less. Applications which require a larger heap should use 64-bit IBM Technology for Java or the Classic VM. Since applications will use less memory when running in the 32-bit VM, this means that applications which require up to about 5 GB in the Classic VM will probably be able to run in the 32-bit VM. Of course, applications with heap requirements near the 3 GB limit will require extra testing to ensure that they are able to run properly under full load over an extended period of time.
3. Applications which use a single VM to fully utilize large systems (especially 8-way and above) will tend to require larger heap sizes, and therefore may not be able to use the 32-bit VM. In some cases it may be possible to divide the work across two or more VMs. Otherwise, it may be necessary to use one of the 64-bit VMs on large systems to allow larger heap sizes.
4. Because calls to native ILE code are more expensive in IBM Technology for Java, extra care should be taken when moving Java applications which make heavy use of native ILE code to the new VM. Performance testing should be performed to determine whether or not the overhead of the native ILE calls are hurting performance in your application. If this is an issue, the techniques discussed above should be used to attempt to improve the performance. If the performance is still unacceptable, it may be best to continue using the Classic VM at this time. Conversely, applications which make use of i5/OS PASE native methods may see a performance improvement when running in IBM Technology for Java due to the reduced overhead of calling i5/OS PASE methods.
5. Remember that microbenchmarks (small tests to exercise a specific function) do not provide a good measure of performance. Comparisons between the IBM Technology for Java and Classic based on microbenchmarks will not give an accurate picture of how your application will perform in the two VMs, because your application will have different characteristics than the microbenchmark. The best way to determine which VM provides the best performance for your application is to test with the

application itself or a reasonably complete subset of the application, using a load generating tool to simulate a load representative of your planned deployment environment.

WebSphere applications running with IBM Technology for Java will be subject to the same constraints as plain Java applications; however, there are some considerations which are specific to WebSphere, as described in Chapter 6 (Web Server and WebSphere Performance).

## 7.6 Capacity Planning

Due to the wide variety of Java applications which can be developed, it is impossible to make precise capacity planning recommendations which would apply to all applications. It is possible, however, to make some general statements which will apply to most applications. Determining specific system requirements for a particular application requires performance testing with that application. The Workload Estimator can also be used to assist with capacity planning for specific environments, such as WebSphere Application Server or WebSphere Commerce applications.

Despite substantial progress at the language execution level, Java continues to require, on average, processors with substantially higher capabilities than the same machine primarily running RPG and COBOL. This is partially due to the overhead of using an object oriented, garbage collected language. But perhaps more important is that Java applications tend to do more than their counterparts written in more traditional languages. For example, Java applications frequently include more network access and data transformation (like XML) than the RPG and COBOL applications they replace. Java applications also typically use JDBC with SQL to access the database, while traditional iSeries applications tend to use less expensive data access methods like Record Level Access. Therefore, Java applications will continue to require more processor cycles than applications with "similar" functionality written in RPG or COBOL.

As a result, some models at the low end may be suitable for traditional applications, but will not provide acceptable performance for applications written in Java.

**General Guidelines**

- Remember to account for non-Java work on the system. Few System i servers are used for a single application; most will have a combination of Java and non-Java applications running on the system. Be sure to factor in capacity requirements for both the Java and the non-Java applications which will run on the system. The eServer Workload Estimator can be used to estimate system requirements for a variety of application types.

- Similarly, be sure to consider additional system services which will be used when adding a new Java application to the system. Most Java applications will make use of system services like network communications and database, which may require additional system resources. In particular, the use of JDBC and dynamic SQL can increase the cost of database access from Java compared to traditional applications with similar function.

- Also consider which applications on the system are likely to experience future growth, and adjust the system requirements accordingly. For example, if a Java/WebSphere application is used as the core of an e-business application, then it may see significantly more growth (requiring additional system resources) over time or during particular times of the year than other applications on the system.

- Beware of misleading benchmarks. Many benchmarks are available to test Java performance, but most of these are not good predictors of server-side Java performance. Some of these benchmarks are single-threaded, or run for a very short period of time. Others will stress certain components of the JVM heavily, while avoiding other functionality that is more typical of real applications. Even the best benchmarks will exercise the JVM differently than real applications with real data. This doesn't mean that benchmarks aren't useful; however, results from these benchmarks must be interpreted carefully.

- 5250 OLTP isn't needed for Java applications, although some Java applications will execute 5250 operations that do require 5250 OLTP. Again, be sure to account for non-Java workloads on the system that do require 5250 OLTP.

- Java applications are inherently multi-threaded. Even if the application itself runs in a single thread, VM functionality like Garbage Collection and asynchronous JIT compilation will run in separate threads. As a result, Java will tend to benefit from processors which support Simultaneous Multi-threading (SMT). See Chapter 20 for additional information on SMT. Java applications may also benefit more from systems with multiple processors than single-threaded traditional applications, as multiple application threads can be running in parallel.

- Java tends to require more main storage (memory) than other languages, especially when using the Classic VM. The 64-bit VMs (both Classic and IBM Technology for Java) will also tend to require more memory than is needed by 32-bit VMs on other platforms.

- Along the same lines, Java applications generally benefit more from L3 cache than applications in other languages. Therefore, Java performance may scale better than CPW ratings would indicate when moving from a system with no L3 cache to a system that does have L3 cache. Conversely, Java performance on a system without L3 cache may be worse than the CPW rating suggests. See Appendix C of this document for information on which systems include L3 cache.

- DASD (hard disk) requirements typically don't change much for Java applications compared to applications written in languages like RPG. The biggest use of DASD is usually database, and database sizes do not inherently change when running Java.

---

## 7.7 Java Performance – Tips and Techniques

### Introduction

Tips and techniques for Java fall into several basic categories:

1. i5/OS Specific. These should be checked out first to ensure you are getting all you should be from your i5/OS Java application.

2. Classic VM Specific. Many i5/OS-specific tips apply only when using the Classic VM and not for IBM Technology for Java.

3. Java Language Specific. Coding tips that will ordinarily improve any Java application, or especially improve it on i5/OS.

4. Database Specific. Use of database can invoke significant path length in i5/OS. Invoking it efficiently can maximize the performance and value of a Java application.

### i5/OS Specific Java Tips and Techniques

- *Load the latest CUM package and PTFs*
  To be sure that you have the best performing code, be sure to load the latest CUM packages and PTFs for all products that you are using. In particular, performance improvements are often introduced in new Java Group PTFs (SF99269 for V5R3, SF99291 for V5R4, and SF99562 for V6R1).

- *Explore the General Performance Tips and Techniques in Chapter 20*
  Some of the discussion in that chapter will apply to Java. Pay particular attention to the discussion "Adjusting Your Performance Tuning for Threads." Specifically, ensure that MAXACT is set high enough to allow all Java threads to run.

- *Consider running Java applications in a separate memory pool*
  On systems running multiple workloads simultaneously, putting Java applications in their own pool will ensure that the Java applications have enough memory allocated to them.

- *Make sure SMT is enabled on systems that support it*
  Java applications are always multi-threaded, and should benefit from Simultaneous Multi-threading (SMT). Ensure that it is turned on by setting the system value QPRCMLTTSK to 1 (On). See chapter 20 for additional details on SMT.

- *Avoid starting new Java VMs frequently*
  Starting a new VM (e.g. through the JAVA/RUNJVA commands) is expensive on any platform, but perhaps a bit more so on i5/OS, due to the relatively high cost of starting a new job. Other factors which make Java startup slow include class loading, bytecode verification, and JIT compilation. As a result, it is far better to use long-running Java programs rather than frequently starting new VMs. If you need to invoke Java frequently from non-Java programs, consider passing messages through an i5/OS Data Queue. The ToolBox Data Queue classes may be used to implement "hot" VM's.

### Classic VM-specific Tips

- *Use java.compiler=jitc*
  The JIT compiler now outperforms Direct Execution for nearly all applications. Therefore, java.compiler=jitc should be used for most Java applications. One possible exception is when startup time is a critical issue, but JIT may be appropriate even in these cases. Setting `java.compiler` is not necessary for Classic on V6R1, or for IBM Technology for Java on either V5R4 or V6R1 -- the JIT compiler is always used in these cases.

- *Delete existing DE program objects*
  When using the JIT, JVAPGM objects containing Direct Execution machine code are not used. These program objects can be large, so removing the unused JVAPGM objects can free up disk space. This is not needed on V6R1. To determine if your class/zip/jar file has a permanent, hidden program object on previous releases, use the DSPJVAPGM command. If a Java program is associated with the file, and the “Optimization” level is something other than *INTERPRET, use DLTJVAPGM to delete the hidden program. DLTJVAPGM does not affect the jar or zip file itself; only the hidden program. Do not use DLTJVAPGM on IBM-shipped JDK jar files (such as rt.jar). As explained earlier, the JIT

does take advantage of programs created at optimization *INTERPRET. These programs require significantly less space and do not need to be deleted. Program objects (even at *INTERPRET) are not used by IBM Technology for Java.

- *Consider the special property os400.jit.mmi.threshold.*
  This property sets the threshold for the MMI of the JIT. Setting this to a small value will result is compilation of the classes at startup time and will increase the start up time. In addition, using a very small value (less than 50) may result in a slower compiled version, since profiling data gathered during the interpreted phase may not be representative of the actual application characteristics. Setting this to a high value may result in a somewhat faster startup time and compilation of the classes will occur once the threshold is reached. However, if the value is set too high then an increased warm-up time may occur since it will take additional time for the classes to be optimized by the JIT compiler.

  The default value of 2000 is usually OK for most scenarios. This property has no effect when using IBM Technology for Java.

- *Package your Java application as a .jar or .zip file.*
  Packaging multiple classes in one .zip or .jar file should improve class loading time and also code optimization when using Direct Execution (DE). Within a .zip or .class file, i5/OS Java will attempt to in-line code from other members of the .zip or .jar file.

**Java Language Performance Tips**

Due to advances in JIT technology, many common code optimizations which were critical for performance a few years ago are no longer as necessary in modern JVMs. Even today, these techniques will not hurt performance. But they may not make a big positive difference either. When making these types of optimizations, care should be taken to balance the need for performance with other factors such as code readability and the ease of future maintenance. It is also important to remember that the majority of the application's CPU time will be spent in a small amount of code. CPU profiling should be used to identify these "hot spots", and optimizations should be focused on these sections of code.

Various Java code optimizations are well documented. Some of the more common optimizations are described below:

- *Minimize object creation*
  Excessive object creation is a common cause of poor application performance. In addition to the cost of allocating memory for the new object and invoking its constructor, the new object will use space in the Java heap, which will result in longer garbage collection cycles. Of course, object creation cannot be avoided, but it can be minimized in key areas.

  The most important areas to look at for reducing object creation is inside loops and other commonly-executed code paths. Some common causes of object creation include:

  - *String.substring*( ) creates a new String object.

  - The arithmetic methods in java.math.BigDecimal (*add*, *divide*, etc) create a new BigDecimal object.

- The I/O method *readLine*( ) (e.g. in java.io.BufferedReader) will create a new String.

- String concatenation (e.g.: "The value is: " + value) will generally result in creation of a StringBuffer, a String, and a character array.

- Putting primitive values (like *int* or *long*) into a collection (like *List* or *Map*) requires wrapping it in a new object (e.g. Java.lang.Integer). This is usually obvious in the code, but Java 5.0 introduced the concept of *autoboxing* which will perform this wrapping automatically, hiding the object creation from the programmer.

Some objects, like StringBuffer, provide a way to reset the object to its initial state, which can be useful for avoiding object creation, especially inside loops. For StringBuffer, this can be done by calling *setLength(0)*.

- *Minimize synchronized methods*
  Synchronized methods/blocks can have significantly more overhead than non-synchronized code. This includes some overhead in acquiring locks, flushing caches to correctly implement the Java memory model, and contention on locks when multiple threads are trying to hold the same lock at the same time. From a performance standpoint, it is best if synchronized code can be avoided. However, it is important to remember that improperly synchronized code can cause functional or data-integrity issues; some of these issues may be difficult to debug since they may only occur under heavy load. As a result, it is important to ensure that changes to synchronization are "safe". In many cases, removing synchronization from code may require design changes in the application.

  Some common synchronization patterns are easily illustrated with Java's built-in String classes. Most other Java classes (including user-written classes) will follow one of these patterns. Each has different performance characteristics.

  - java.lang.String is an *immutable* object – once constructed, it cannot be changed. As a result, it is inherently thread-safe and does not require synchronization. However, since Strings cannot be modified, operations which require a modified String (like *String.substring()*) will have to create a new String, resulting in more object creation.

  - java.lang.StringBuffer is a mutable object which can change after it is constructed. In order to make it thread-safe, nearly all methods in the class (including some which do not modify the StringBuffer) are synchronized.

  - java.lang.StringBuilder (introduced in Java 5.0) is an unsynchronized version of StringBuffer. Because its methods are not synchronized, this class is not thread-safe, so StringBuilder instances can not be shared between threads without external synchronization.

  Dealing with synchronization correctly requires a good understanding of Java and your application, so be careful about applying this tip.

- *Use exceptions only for "exceptional" conditions*
  The "try" block of an exception handler carries little overhead. However, there is significant overhead when an exception is actually thrown and caught. Therefore, you should use exceptions only for "exceptional" conditions; that is, for conditions that are not likely to happen during normal execution. For example, consider the following procedure:

```
public void badPrintArray (int arr[]) {
```

```
    int i = 0;
    try {
        while (true) {
            System.out.println (arr[i++]);
        }
    } catch (ArrayOutOfBoundsException e) {
        // Reached the end of the array....exit
    }
}
```

Instead, the above procedure should be written as:

```
public void goodPrintArray (int arr[]) {
    int len = arr.length;
    for (int i = 0; i < len; i++) {
        System.out.println (arr[i]);
    }
}
```

In the “bad” version of this code, an exception will always be thrown (and caught) in every execution of the method. In the “good” version, most calls to the method will not result in an exception. However, if you passed “null” to the method, it would throw a NullPointerException. Since this is probably not something that would normally happen, an exception may be appropriate in this case. (On the other hand, if you expect that null will be passed to this method frequently, it may be a good idea to handle it specifically rather than throwing an exception.)

- *Use static final when creating constants*
  When data is invariant, declare it as static final. For example here are two array initializations:

```
class test1 {
      int myarray[] =
                { 1,2,3,4,5,6,7,8,9,10,
                  2,3,4,5,6,7,8,9,10,11,
                  3,4,5,6,7,8,9,10,11,12,
                  4,5,6,7,8,9,10,11,12,13,
                  5,6,7,8,9,10,11,12,13,14 };
}

class test2 {
      static final int myarray2[] =
                  { 1,2,3,4,5,6,7,8,9,10,
                    2,3,4,5,6,7,8,9,10,11,
                    3,4,5,6,7,8,9,10,11,12,
                    4,5,6,7,8,9,10,11,12,13,
                    5,6,7,8,9,10,11,12,13,14 };
}
```

Since the array myarray2 in class test2 is defined as *static*, there is only one myarray2 array for all the many creations of the test2 object. In the case of the test1 class, there is an array myarray for *each* test1 instance. The use of *final* ensures that the array cannot be changed, making it safe to use from multiple threads.

### Java i5/OS Database Access Tips

- *Use the native JDBC driver*
  There are two i5/OS JDBC drivers that may be used to access local data: the Native driver (using a JDBC URL *"jdbc:db2:system-name"*) and the Toolbox driver (with a JDBC URL *"jdbc:as400:system-name"*). The native JDBC driver is optimized for local database access, and gives the best performance when accessing the database on the same system as your Java

applications. The Toolbox driver supports remote access, and should be used when accessing the database on a separate system. This recommendation is true for both the 64-bit Classic VM and the new 32-bit VM.

- *Pool Database Connections*
  Connection pooling is a technique for sharing a small number of database connections among a number of threads. Rather than each thread opening a connection to the database, executing some requests, and then closing the connection, a connection can be obtained from the connection pool, used, and then returned to the pool. This eliminates much of the overhead in establishing a new JDBC connection. WebSphere Application Server uses built-in connection pooling when getting a JDBC connection from a DataSource.

- *Use Prepared Statements*
  The JDBC *prepareStatement* method should be used for repeatable *executeQuery* or *executeUpdate* methods. If *prepareStatement*, which generates a reusable PreparedStatement object, is not used, the *execute* statement will implicitly re-do this work on every *execute* or *executeQuery*, even if the query is identical. WebSphere's DataSource will automatically cache your PreparedStatements, so you don't have to keep a reference to them – when WebSphere sees that you are attempting to prepare a statement that it has already prepared, it will give you a reference to the already prepared statement, rather than creating a new one. In non-WebSphere applications, it may be necessary to explicitly cache PreparedStatement objects.

  When using PreparedStatements, be sure to use parameter markers for variable data, rather than dynamically building query strings with literal data. This will enable reuse of the PreparedStatement with new parameter values.

  Avoid placing the prepareStatement inside of loops (e.g. just before the execute). In some non-i5/OS environments, this just-before-the-query coding practice is common for non-Java languages, which required a "prepare" function for any SQL statement. Programmers may carry this practice over to Java. However, in many cases, the prepareStatement contents don't change (this includes parameter markers) and the Java code will run faster on all platforms if it is executed only one time, instead of once per loop. This technique may show a greater improvement on i5/OS.

- *Store or at least fetch numeric data in DB2 as double*
  Fixed-precision decimal data cannot be represented in Java as a primitive type. When accessing numeric and decimal fields from the database through JDBC, values can be retrieved using getDouble( ) or getBigDecimal( ). The latter method will create a new java.math.BigDecimal object each time it is called. Using getDouble (which returns a primitive double) will give better performance, and should be preferred when floating-point values are appropriate for your application (i.e. for most applications outside the financial industry).

- *Consider using ToolBox record I/O*
  The IBM Toolbox for Java provides native record level access classes. These classes are specific to the i5/OS platform. They may provide a significant performance gain over the use of JDBC access for applications where portability to other databases is not required. See the AS400File object under Record Level access in the InfoCenter.

## Resources

The i5/OS Java and WebSphere performance team maintains a list of performance-related documents at http://www.ibm.com/systems/i/solutions/perfmgmt/webjtune.html.

The Java Diagnostics Guide provides detailed information on performance tuning and analysis when using IBM Technology for Java. Most of the document applies to all platforms using IBM's Java VM; in addition, one chapter is written specifically for i5/OS information. The Diagnostics Guide is available at http://www.ibm.com/developerworks/java/jdk/diagnosis/.

# Chapter 8. Cryptography Performance

With an increasing demand for security in today's information society, cryptography enables us to encrypt the communication and storage of secret or confidential data. This also requires data integrity, authentication and transaction non-repudiation. Together, cryptographic algorithms, shared/symmetric keys and public/private keys provide the mechanisms to support all of these requirements. This chapter focuses on the way that System i cryptographic solutions improve the performance of secure e-Business transactions.

There are many factors that affect System i performance in a cryptographic environment. This chapter discusses some of the common factors and offers guidance on how to achieve the best possible performance. Much of the information in this chapter was obtained as a result of analysis experience within the Rochester development laboratory. Many of the performance claims are based on supporting performance measurement and other performance workloads. In some cases, the actual performance data is included here to reinforce the performance claims and to demonstrate capacity characteristics.

**Cryptography Performance Highlights for i5/OS V5R4M0:**

- Support for the 4764 Cryptographic Coprocessor is added. This adapter provides both cryptographic coprocessor and secure-key cryptographic accelerator function in a single PCI-X card.
- 5722-AC3 Cryptographic Access Provider withdrawn. This product is no longer required to enable data encryption.
- Cryptographic Services API function added. Key management function has been added, which helps you securely store and handle cryptographic keys.

## 8.1 System i Cryptographic Solutions

On a System i, cryptographic solutions are based on software and hardware Cryptographic Service Providers (CSP). These solutions include services required for Network Authentication Service, SSL/TLS, VPN/IPSec, LDAP and SQL.

**IBM Software Solutions**

The software solutions are either part of the i5/OS Licensed Internal Code or the Java Cryptography Extension (JCE).

**IBM Hardware Solutions**

One of the hardware based cryptographic offload solutions for the System i is the **IBM 4764 PCI-X Cryptography Coprocessor (Feature Code 4806).** This solution will offload portions of cryptographic processing from the host CPU. The host CPU issues requests to the coprocessor hardware. The hardware then executes the cryptographic function and returns the results to the host CPU. Because this hardware based solution handles selected compute-intensive functions, the host CPU is available to support other system activity. SSL/TLS network communications can use these options to dramatically offload cryptographic processing related to establishing an SSL/TLS session.

**CSP API Sets**

User applications can utilize cryptographic services indirectly via i5/OS functions (SSL/TLS, VPN IPSec) or directly via the following APIs:

- The Common Cryptographic Architecture (CCA) API set is provided for running cryptographic operations on a Cryptographic Coprocessor.
- The i5/OS Cryptographic Services API set is provided for running cryptographic operations within the Licensed Internal Code.
- Java Cryptography Extension (JCE) is a standard extension to the Java Software Development Kit (JDK).
- GSS (Generic Security Services), Java GSS, and Kerberos APIs are part of the Network Authentication Service that provides authentication and security services. These services include session level encryption capability.
- i5/OS SSL and JSSE support the Secure Sockets Layer Protocol. APIs provide session level encryption capability.
- Structured Query Language is used to access or modify information in a database. SQL supports encryption/decryption of database fields.

## 8.2 Cryptography Performance Test Environment

All measurements were completed on an IBM System i5 570+ 8-Way (2.2 GHz). The system is configured as an LPAR, and each test was performed on a single partition with one dedicated CPU. The partition was solely dedicated to run each test. The IBM 4764 PCI-X Cryptographic Coprocessor card is installed in a PCI-X slot.

This System i model is a POWER5 hardware system, which provides Simultaneous Multi-Threading. The tools used to obtain this data are in some cases only single threaded (single instruction stream) applications, which don't take advantage of the performance benefits of SMT. See section 8.6 for additional information.

Cryptperf is an IBM internal use primative-level cryptographic function test driver used to explore and measure System i cryptographic performance. It supports parameterized calls to various i5/OS CSPs. See section 8.6 for additional information.

- **Cipher:** Measures the performance of either symmetric or asymmetric key encrypt depending on algorithm selected.
- **Digest:** Measures the performance of hash functions.
- **Sign:** Measures the performance of hash with private key encrypt .
- **Pin:** Measures encrypted PIN verify using the IBM 3624 PIN format with the IBM 3624 PIN calculation method.

All i5/OS and JCE test cases run at a near 100% CPU utilization. The test cases that use the Cryptographic Coprocessor will offload all cryptographic functions, so that CPU utilization is negligible.

The relative performance and recommendations found in this chapter are similar for other models, but the data presented here is not representative of a specific customer environment. Cryptographic functions are very CPU intensive and scale easily. Adding or removing CPU's to an environment will change performance, so results in other environments may vary significantly.

## 8.3 Software Cryptographic API Performance

This section provides performance information for System i systems using the following cryptographic services; i5/OS Cryptographic Services API and IBM JCE 1.2.1, an extension of JDK 1.4.2.

Cryptographic performance is an important aspect of capacity planning, particularly for applications using secure network communications. The information in this section may be used to assist in capacity planning for this complex environment.

### Measurement Results

The cryptographic performance measurements in the following three tables were made using i5/OS Cryptographic Services API and Java Cryptography Extension.

*Table 8.1*

**Cipher Encrypt Performance**

| Encryption Algorithm | Threads | Key Length (Bits) | Transaction Length (Bytes) | i5/OS (Transactions/ Second) | i5/OS (Bytes/ Second) | JCE (Transactions /Second) | JCE (Bytes/ Second) |
|---|---|---|---|---|---|---|---|
| DES | 1 | 56 | 1024 | 11,276 | 11,547,058 | 15,537 | 15,909,515 |
| DES | 10 | 56 | 1024 | 15,402 | 15,771,656 | 19,768 | 20,241,955 |
| Triple DES | 1 | 112 | 1024 | 5,039 | 5,159,756 | 5,997 | 6,140,893 |
| Triple DES | 1 | 112 | 65536 | 87 | 5,710,925 | 93 | 6,086,464 |
| Triple DES | 10 | 112 | 1024 | 6,625 | 6,783,658 | 7,517 | 7,697,917 |
| Triple DES | 10 | 112 | 65536 | 109 | 7,139,814 | 117 | 7,657,551 |
| RC4 | 1 | 128 | 262144 | 947 | 248,224,207 | 125 | 32,704,635 |
| RC4 | 10 | 128 | 262144 | 1,017 | 266,579,889 | 207 | 54,321,919 |
| AES | 1 | 128 | 1024 | 26,636 | 27,275,585 | 28,110 | 28,784,259 |
| AES | 1 | 128 | 65536 | 1,479 | 96,930,853 | 428 | 28,080,038 |
| AES | 1 | 256 | 1024 | 24,025 | 24,601,428 | 22,767 | 23,313,526 |
| AES | 1 | 256 | 65536 | 1,111 | 72,782,397 | 345 | 22,614,607 |
| AES | 10 | 128 | 1024 | 30,408 | 31,137,523 | 34,916 | 35,754,190 |
| AES | 10 | 128 | 65536 | 1,692 | 110,892,831 | 524 | 34,350,709 |
| AES | 10 | 256 | 1024 | 27,349 | 28,005,446 | 27,172 | 27,824,575 |
| AES | 10 | 256 | 65536 | 1,257 | 82,392,038 | 415 | 27,183,773 |
| RSA | 1 | 1024 | 100 | 897 | n/a | 197 | n/a |
| RSA | 1 | 2048 | 100 | 128 | n/a | 30 | n/a |
| RSA | 10 | 1024 | 100 | 1,187 | n/a | 246 | n/a |
| RSA | 10 | 2048 | 100 | 165 | n/a | 35 | n/a |

**Notes:**
- See section 8.2 for Test Environment Information

*Table 8.2*

**Signing Performance**

| Encryption Algorithm | Threads | RSA Key Length (Bits) | i5/OS (Transactions/Second) | JCE (Transactions/Second) |
|---|---|---|---|---|
| SHA-1 / RSA | 1 | 1024 | 901 | 197 |
| SHA-1 / RSA | 10 | 1024 | 1,155 | 240 |
| SHA-1 / RSA | 1 | 2048 | 129 | 30 |
| SHA-1 / RSA | 10 | 2048 | 163 | 35 |

**Notes:**
- Transaction Length set at 1024 bytes
- See section 8.2 for Test Environment Information

*Table 8.3*

**Digest Performance**

| Encryption Algorithm | Threads | i5/OS (Transactions/ Second) | i5/OS (Bytes/ Second) | JCE (Transactions/ Second) | JCE (Bytes/Second) |
|---|---|---|---|---|---|
| SHA-1 | 1 | 6,753 | 110,642,896 | 2,295 | 37,608,172 |
| SHA-1 | 10 | 10,875 | 178,172,751 | 2,954 | 48,401,773 |
| SHA-256 | 1 | 3,885 | 63,645,228 | 2,049 | 33,576,523 |
| SHA-256 | 10 | 4,461 | 73,086,411 | 2,392 | 39,184,923 |
| SHA-384 | 1 | 7,050 | 115,505,548 | 4,020 | 65,865,327 |
| SHA-384 | 10 | 8,075 | 132,301,878 | 4,634 | 75,925,668 |
| SHA-512 | 1 | 7,031 | 115,201,800 | 4,217 | 69,098,731 |
| SHA-512 | 10 | 8,060 | 132,059,807 | 4,801 | 78,659,561 |

**Notes:**
- Key Length set at 1024 bits
- Transaction Length set at 16384 bytes
- See section 8.2 for Test Environment Information

## 8.4 Hardware Cryptographic API Performance

This section provides information on the hardware based cryptographic offload solution **IBM 4764 PCI-X Cryptography Coprocessor (Feature Code 4806)**. This solution will improve the system CPU capacity by offloading CPU demanding cryptographic functions.

| | |
|---|---|
| **IBM Common Name** | **IBM 4764 PCI-X Cryptographic Coprocessor** |
| **System i hardware feature code** | #4806 |
| **Applications** | Banking/finance (B/F) |
| | Secure accelerator (SSL) |
| **Cryptographic Key Protection** | Secure hardware module |
| **Required Hardware** | No IOP Required |
| **Platform Support** | IBM System i5 |

The 4764 Cryptographic Coprocessor provides both cryptographic coprocessor and secure-key cryptographic accelerator functions in a single PCI-X card. The coprocessor functions are targeted to banking and finance applications. The secure-key accelerator functions are targeted to improving the performance of SSL (secure socket layer) and TLS (transport layer security) based transactions. The 4764 Cryptographic Coprocessor supports secure storage of cryptographic keys in a tamper-resistant module,

which is designed to meet FIPS 140-2 Level 4 security requirements. This new cryptographic card offers the security and performance required to support e-Business and emerging digital signature applications.

For banking and finance applications the 4764 Cryptographic Coprocessor delivers improved performance for T-DES, RSA, and financial PIN processing. IBM CCA (Common Cryptographic Architecture) APIs are provided to enable finance and other specialized applications to access the services of the coprocessor. For banking and finance applications the 4764 Coprocessor is a replacement for the 4758-023 Cryptographic Coprocessor (feature code 4801).

The 4764 Cryptographic Coprocessor can also be used to improve the performance of high-transaction-rate secure applications that use the SSL and TLS protocols. These protocols are used between server and client applications over a public network like the Internet, when private information is being transmitted in the case of Consumer-to-Business transactions (for example, a web transaction with payment information containing credit card numbers) or Business-to-Business transactions. SSL/TLS is the predominant method for securing web transactions. Establishing SSL/TLS secure web connections requires very compute intensive cryptographic processing. The 4764 Cryptographic Coprocessor off-loads cryptographic RSA processing associated with the establishment of a SSL/TLS session, thus freeing the server for other processing. For cryptographic accelerator applications the 4764 Cryptographic Coprocessor is a replacement for the 2058 Cryptographic Accelerator (feature code 4805).

Cryptographic performance is an important aspect of capacity planning, particularly for applications using SSL/TLS network communications. Besides host processing capacity, the impact of one or more Cryptographic Coprocessors must be considered. Adding a Cryptographic Coprocessor to your environment can often be more beneficial then adding a CPU. The information in this chapter may be used to assist in capacity planning for this complex environment.

### Measurement Results

The following three tables display the cryptographic test cases that use the Common Cryptographic Architecture (CCA) interface to measure transactions per second for a variety of 4764 Cryptographic Coprocessor functions.

*Table 8.4*

**Cipher Encrypt Performance**
**CCA CSP**

| Encryption Algorithm | Threads | Key Length (Bits) | Transaction Length (Bytes) | 4764 (Transactions/second) | 4764 (Bytes/second) |
|---|---|---|---|---|---|
| DES | 1 | 56 | 1024 | 1,026 | 1,050,283 |
| DES | 10 | 56 | 1024 | 1,053 | 1,078,458 |
| Triple DES | 1 | 112 | 1024 | 1,002 | 1,025,798 |
| Triple DES | 1 | 112 | 65536 | 110 | 7,191,327 |
| Triple DES | 10 | 112 | 1024 | 1,021 | 1,045,535 |
| Triple DES | 10 | 112 | 65536 | 123 | 8,035,164 |
| RSA | 1 | 1024 | 100 | 796 | n/a |
| RSA | 1 | 2048 | 100 | 307 | n/a |
| RSA | 10 | 1024 | 100 | 1,044 | n/a |
| RSA | 10 | 2048 | 100 | 462 | n/a |

**Notes:**
- See section 8.2 for Test Environment information
- AES is not supported by the IBM 4764 Cryptographic Coprocessor

*Table 8.5*

**Signing Performance**
**CCA CSP**

| Encryption Algorithm | Threads | RSA Key Length (Bits) | 4764 (Transactions/second) |
|---|---|---|---|
| SHA-1 / RSA | 1 | 1024 | 794 |
| SHA-1 / RSA | 10 | 1024 | 1,074 |
| SHA-1 / RSA | 1 | 2048 | 308 |
| SHA-1 / RSA | 10 | 2048 | 465 |

**Notes:**
- Transaction Length set at 1024 bytes
- See section 8.2 for Test Environment information

*Table 8.6*

**Financial PINs Performance**
**CCA CSP**

| Threads | Total Repetitions | 4764 (Transactions/second) |
|---|---|---|
| 1 | 10000 | 945 |
| 10 | 100000 | 966 |

**Notes:**
- See section 8.2 for Test Environment information

## 8.5 Cryptography Observations, Tips and Recommendations

- The IBM Systems Workload Estimator, described in Chapter 23, reflects the performance of real user applications while averaging the impact of the differences between the various communications protocols. The real world perspective offered by the Workload Estimator may be valuable in some cases

- SSL/TLS client authentication requested by the server is quite expensive in terms of CPU and should be requested only when needed. Client authentication full handshakes use two to three times the CPU resource of server-only authentication. RSA authentication requests can be offloaded to an IBM 4764 Cryptographic Coprocessor.

- With the use of Collection Services you can count the SSL/TLS handshake operations. This capability allows you to better understand the performance impact of secure communications traffic. Use this tool to count how many full versus cached handshakes per second are being serviced by the server. Start the Collection Services with the default “Standard plus protocol”. When the collection is done you can find the SSL/TLS information in the QAPMJOBMI database file in the fields JBASH (full) and JBFSHA (cached) for server authentications or JBFSHA (full) and JBASHA (cached) for server and client authentications. Accumulate the full handshake numbers for all jobs and you will have a good method to determine the need for a 4764 Cryptographic Coprocessor. Information about Collection Services can be found at the System i Information Center. See section 8.6 for additional information.

- Symmetric key encryption and signing performance improves significantly when multithreaded.

- Supported number of 4764 Cryptographic Coprocessors:

*Table 8.8*

| server models | Maximum per server | Maximum per partition |
|---|---|---|
| IBM System i5 570 8/12/16W, 595 | 32 | 8 |
| IBM System i5 520, 550, 570 2/4W | 8 | 8 |

- Applications requiring a FIPS 140-2 Level 4 certified, tamper resistant module for storing cryptographic keys should use the IBM 4764 Cryptographic Coprocessor.

- Cryptographic functions demand a lot of a system CPU, but the performance does scale well when you add a CPU to your system. If your CPU handles a large number of cryptographic requests, offloading them to an IBM 4764 Cryptographic Coprocessor might be more beneficial then adding a new CPU.

## 8.6 Additional Information

Extensive information about using System i Cryptographic functions may be found under “Security” and “Networking Security” at the System i Information Center web site at:
http://www.ibm.com/eserver/iseries/infocenter .

IBM Security and Privacy specialists work with customers to assess, plan, design, implement and manage a security-rich environment for your online applications and transactions. These Security, Privacy, Wireless Security and PKI services are intended to help customers build trusted electronic relationships with employees, customers and business partners. These general IBM security services are described at:
http://www.ibm.com/services/security/index.html .

General security news and information are available at: http://www.ibm.com/security .

System i Security White Paper, “Security is fundamental to the success of doing e-business” is available at:
http://www.ibm.com/security/library/wp_secfund.shtml .

IBM Global Services provides a variety of Security Services for customers and Business Partners. Their services are described at: http://www.ibm.com/services/ .

Links to other Cryptographic Coprocessor documents including custom programming information can be found at: http://www.ibm.com/security/cryptocards .

Other performance information can be found at the System i Performance Management website at:
http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.html .

More details about POWER5 and SMT can be found in the document *Simultaneous Multi-Threading (SMT) on eServer iSeries POWER5 Processors* at:
http://www.ibm.com/servers/eserver/iseries/perfmgmt/pdf/SMT.pdf .

# Chapter 9. iSeries NetServer File Serving Performance

This chapter will focus on iSeries NetServer File Serving Performance.

## 9.1 iSeries NetServer File Serving Performance

iSeries Support for Windows Network Neighborhood (iSeries NetServer) supports the Server Message Block (SMB) protocol through the use of Transmission Control Protocol/Internet Protocol (TCP/IP) on iSeries. This communication allows clients to access iSeries shared directory paths and shared output queues. PC clients on the network utilize the file and print-sharing functions that are included in their operating systems. iSeries NetServer properties and the properties of iSeries NetServer file shares and print shares are configured with iSeries Navigator.

Clients can use iSeries NetServer support to install Client Access from the iSeries since the clients use function that is included in their operating system. See: http://www-1.ibm.com/servers/eserver/iseries/netserver for additional information concerning iSeries NetServer.

In **V5R4**, enhancements were made help optimize the performance of the iSeries NetServer, increasing throughput and reducing client response time. The optimizations allow access to thread safe file systems in the integrated file system from a new multithreaded file serving job. In addition, other optimizations have been added and are used when accessing/using files in the "root" (/), QOpenSys, and user-defined file systems (UDFS). See the iSeries NetServer articles in the iSeries Information Center for more information.

***iSeries NetServer Performance***

***Server***
iSeries partition with 2 dedicated processors having equivalent CPW of 2400.
16384 MB main memory
5-4318 CCIN 6718 18 GB disk drives
2-5700 1000 MB (1 GB) Ethernet IOAs[2]

***Client***s
60 6862-27U IBM PC 300PL Pentium II 400 MHz 512KB L2, 320 MB RAM, 6.4 GB disk drive
Intel® 8255x based PCI Ethernet Adapter 10/100
Microsoft Windows XP Professional Version 2002 Service Pack 1

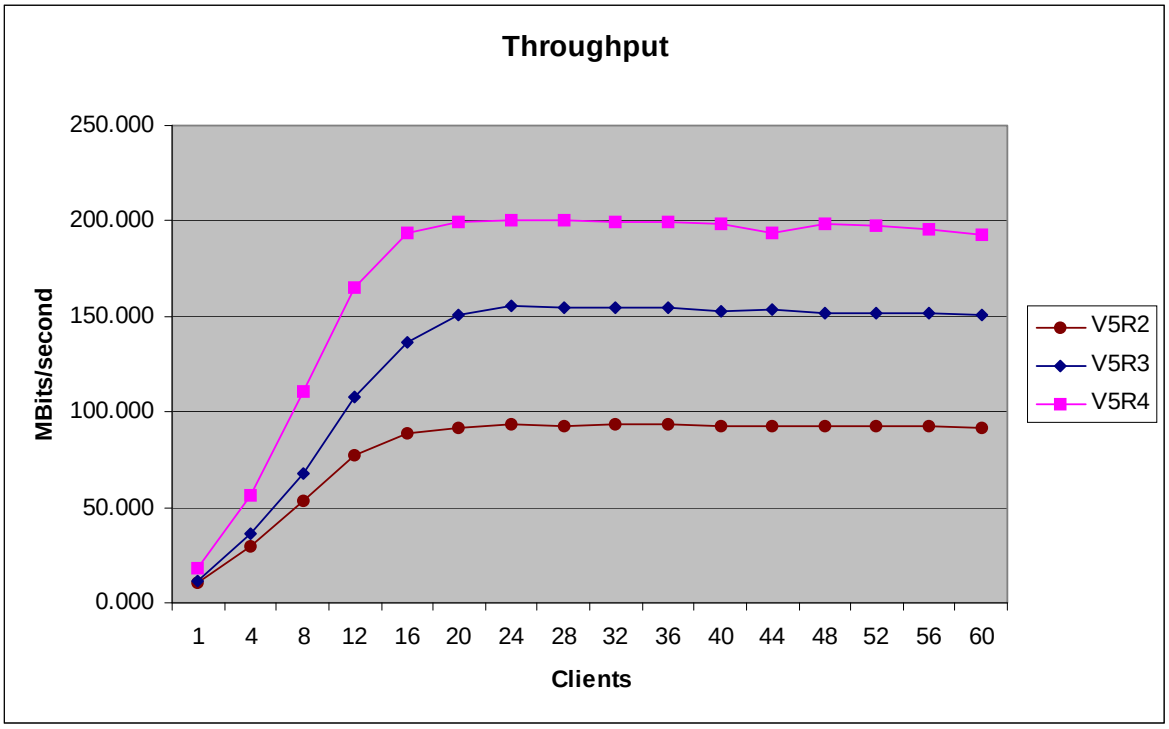
Controller PC: 6862-27U IBM PC 300PL Pentium II 400 MHz 512KB L2, 320 MB RAM, 6.4 GB disk drive Intel® 8255x based PCI Ethernet Adapter 10/100
Microsoft Windows 2000 5.00.2195 Service Pack 4

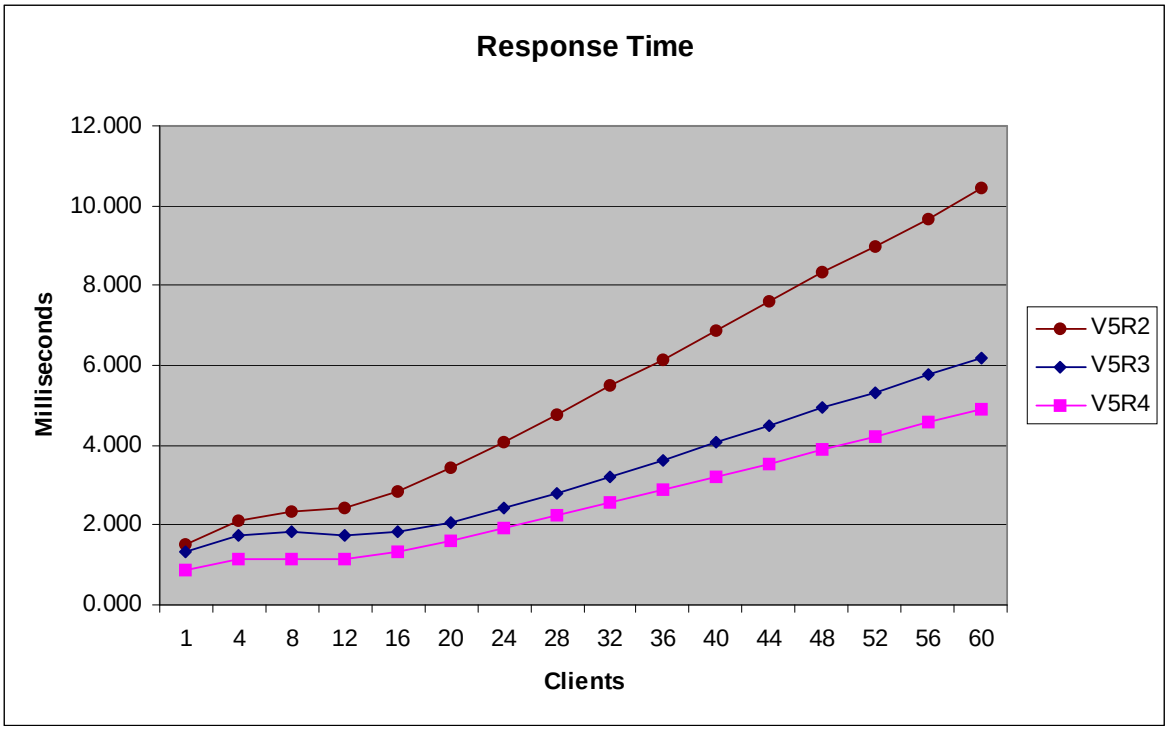
***Workload***
PC Magazine's NetBench® 7.0.3 with the test suite ent_dm.tst was used to provide the benchmark data[3].

[2] The clients used 100 MB Ethernet and were switched into the 1 GB network of the server.

[3] The testing was performed without independent verification by VeriTest testing division of Lionbridge Technologies, Inc. ("VeriTest") or Ziff Davis Media Inc. and that neither Ziff Davis Media Inc. nor VeriTest make any representations or warranties as to the result of the test. NetBench® is a registered trademark of Ziff Davis Media Inc. or its affiliates in the U.S. and other countries. Further details on the test

**Measurement Results**:




## Conclusion/Explanations:

environment can be obtained by sending an email to llhirsch@us.ibm.com.

From the charts above in the Measurement Results section, it is evident that when customers upgrade to V5R4 they can  expect to see an improvement in throughput and response time when using iSeries NetServer.

# Chapter 10. DB2 for i5/OS JDBC and ODBC Performance

DB2 for i5/OS can be accessed through many different interfaces. Among these interfaces are: Windows .NET, OLE DB, Windows database APIs, ODBC and JDBC. This chapter will focus on access through JDBC and ODBC by providing programming and tuning hints as well as links to detailed information.

## 10.1 DB2 for i5/OS access with JDBC

Access to the System i data from portable Java applications can be achieved with the universal database access APIs available in JDBC (Java Database Connectivity). There are two JDBC drivers for the System i. The Native JDBC driver is a type 2 driver. It uses the SQL Call Level Interface for database access and is bundled in the System i Developer Kit for Java. The JDBC Toolbox driver is a type 4 driver which is bundled in the System i Toolbox for Java. In general, the Native driver is chosen when running on the System i server directly, while the Toolbox driver is typically chosen when accessing data on the System i server from another machine. The Toolbox driver is typically used when accessing System i data from a Windows machine, but it could be used when accessing the System i server from any Java capable system. More detailed information on which driver to choose may be found in the JDBC references.

### JDBC Performance Tuning Tips

JDBC performance depends on many factors ranging from generic best programming practices for databases to specific tuning which optimizes JDBC API performance. Tips for both SQL programming and JDBC tuning techniques to improve performance are included here.

- In general when accessing a database it takes less time to retrieve smaller amounts of data. This is even more significant for remote database access where the data is sent over a network to a client. For good performance, SQL queries should be written to retrieve only the data that is needed. Select only needed fields so that additional data is not unnecessarily retrieved and sent. Use appropriate predicates to minimize row selection on the server side to reduce the amount of data sent for client processing.

- Follow the 'Prepare once, execute many times' rule of thumb. For statements that are executed many times, use the PreparedStatement object to prepare the statement once. Then use this object to do subsequent executes of this statement. This significantly reduces the overhead of parsing and compiling the statement every time it is executed.

- Do not use a PreparedStatement object if an SQL statement is run only one time. Compiling and running a statement at the same time has less overhead than compiling the statement and running it in two separate operations.

- Consider using JDBC stored procedures. Stored procedures can help reduce network communication time and traffic which improves response time. Java supports stored procedures via CallableStatement objects.

- Turn off autocommit, if possible. Explicitly manage commits in the application, but do not leave transactions uncommitted for long periods of time.

- Use the lowest isolation level required by the application. Higher isolation levels can reduce performance levels as more locking and synchronization are required. Transaction levels in order of increasing level are: TRANSACTION_NONE, TRANSACTION_READ_UNCOMMITTED, TRANSACTION_READ_COMMITTED, TRANSACTION_REPEATABLE_READ, TRANSACTION_SERIALIZABLE

- Reuse connections. Minimize the opening and closing of connections where possible. These operations are very expensive. If possible, keep connections open and reuse them. A connection pool can help considerably.

- Consider use of Extended Dynamic support. In generally provides better performance by caching the SQL statements in SQL packages on the System i.

- Use appropriate cursor settings. Use a fetch forward only cursor type if the data does not need to be scrollable. Use read only cursors for retrieving data which will not be updated.

- Use block inserts and batch updates.

- Tune connection properties to maximize application performance. The connection properties are explained in the driver documentation. Among the properties are 'block size' and 'data compression' which should be tuned as follows:

    1. Choose the right 'block size" for the application. 'block size' specifies the amount of data to retrieve from the server and cache on the client. For the Toolbox driver 'block size' specifies the transfer size in kilobytes, with 32 as the default. For the native driver 'block size' specifies the number of rows that will be fetched at a time for a result set, with 32 as the default. When larger amounts of data are retrieved a larger block size may help minimize communication time.
    2. The Toolbox driver has a 'data compression' property to enable compressing the data blocks before sending them to the client. This is set to true by default. In general this gives better response time, but may use more CPU.

**References for JDBC**

- The System i Information Center
  Http://publib.boulder.ibm.com/iseries/

- The home page for Java and DB2 for i5/OS
  http://www-03.ibm.com/systems/i/software/db2/javadb2.html

- Sun's JDBC web page
  http://java.sun.com/products/jdbc/

---

## 10.2 DB2 for i5/OS access with ODBC

ODBC (Open Database Connectivity) is a set of API's which provide clients with an open interface to any ODBC supported database. The ODBC APIs are part of System i Access.

In general, the JDBC Performance tuning tips also apply to the performance of ODBC applications:

- Employ efficient SQL programming techniques to minimize the amount of data processed
- Prepared statement reuse to minimize parsing and optimization overhead for frequently run queries
- Use stored procedures when appropriate to bundle processing into fewer database requests
- Consider extended dynamic package support for SQL statement and package caching
- Process data in blocks of multiple rows rather than single records when possible (e.g. Block inserts)

In addition for ODBC performance ensure that each statement has a unique statement handle. Sharing statement handles for multiple sequential SQL statements causes DB2 on i5/OS to do FULL OPEN operations since the database cursor can not be reused. By ensuring that an SQLAllocStmt is done before any SQLPrepare or SQLExecDirect commands, database processing can be optimized. This is especially important when a set of SQL statements are executed in a loop. Ensuring each SQL statement has its own handle reduces the DB2 overhead.

Tools such as ODBC Trace (available through the ODBC Driver Manager) are useful in understanding what ODBC calls are made and what activity occurs as a result. Client application profilers may also be useful in tuning client applications. These are often included in application development toolkits.

**ODBC Performance Settings**

You may be able to further improve the performance of your ODBC application by configuring the ODBC data source through the Data Sources (ODBC) administrator in the Control Panel. Listed below are some of the parameters that can be set to better tune the performance of the System i Access ODBC Driver. The ODBC performance parameters discussed in detail are:

- Prefetch
- ExtendedDynamic
- RecordBlocking
- BlockSizeKB
- LazyClose
- LibraryView

***Prefetch :*** The Prefetch option is a performance enhancement to allow some or all of the rows of a particular ODBC query to be fetched at PREPARE time. We recommend that this setting be turned ON. However, if the client application uses EXTENDED FETCH (SQLExtendedFetch) this option should be turned OFF.

***ExtendedDynamic:*** Extended dynamic support provides a means to "cache" dynamic SQL statements on the System i server. With extended dynamic, information about the SQL statement is saved away in an SQL package object on the System i the first time the statement is run. On subsequent uses of the statement, System i Access ODBC recognizes that the statement has been run before and can skip a significant part of the processing by using the information saved in the SQL package. Statements which are cached include SELECT, positioned UPDATE and DELETE, INSERT with subselect, DECLARE PROCEDURE, and all other statements which contain parameter markers.

All extended dynamic support is application based. This means that each application can have its own configuration for extended dynamic support. Extended dynamic support as a whole is controlled through the use of the ExtendedDynamic option. If this option in not selected, no packages are used. If the option is selected (default) custom settings per application can be configured with the “Custom Settings Per Application” button. When this button is clicked a “Package information for application” window pops up and package library and name fields can be filled in and usage options can be selected.

Packages may be shared by several clients to reduce the number of packages on the System i server. To enable sharing, the default libraries of the clients must be the same and the clients must be running the same application. Extended dynamic support will be deactivated if two clients try to use the same package but have different default libraries. In order to reactivate extended dynamic support, the package should be deleted from the System i and the clients should be assigned different libraries in which to store the package(s).

***Package Usage:*** The default and preferred performance setting enables the ODBC driver to use the package specified and adds statements to the package as they are run. If the package does not exist when a statement is being added, the package is created on the server.

***Considerations for using package support***: It is recommended that if an application has a fixed number of SQL statements in it, a single package be used by all users. An administrator should create the package and run the application to add the statements from the application to the package. Once that is done, configure all users of the package to not add any further statements but to just use the package. Note that for a package to be shared by multiple users each user must have the same default library listed in their ODBC library list. This is set by using the ODBC Administrator.

Multiple users can add to or use a given package at the same time. Keep in mind that as a statement is added to the package, the package is locked. This could cause contention between users and reduce the benefits of using the extended dynamic support.

If the application being used has statements that are generated by the user and are ad hoc in nature, then it is recommended that each user have his own package. Each user can then be configured to add statements to their private package. Either the library name or all but the last 3 characters of the package name can be changed.

***RecordBlocking:*** The RecordBlocking switch allows users to control the conditions under which the driver will retrieve multiple rows (block data) from the System i. The default and preferred performance setting to Use Blocking will enable blocking for everything except SELECT statements containing an explicit "FOR UPDATE OF" clause.

***BlockSizeKB (choices 2 through 512):*** The BlockSizeKB parameter allows users to control the number of rows fetched from the System i per communications flow (send/receive pair). This value represents the client buffer size in Kilobytes and is divided by the size of one row of data to determine the number of rows to fetch from the System i in one request. The primary use of this parameter is to speed up queries that send a lot of data to the client. The default value 32 will perform very well for most queries. If you have the memory available on the client, setting a higher value may improve some queries.

***LazyClose:*** The LazyClose switch allows users to control the way SQLClose commands are handled by the System i Access ODBC Driver. The default and preferred performance setting enables Lazy Close. Enabling LazyClose will delay sending an SQLClose command to the System i until the next ODBC request is sent. If Lazy Close is disabled, a SQLClose command will cause an immediate explicit flow to the System i to perform the close. This option is used to reduce flows to the System i, and is purely a performance enhancing option.

***LibraryView:*** The LibraryView switch allows users to control the way the System i Access ODBC Driver deals with certain catalog requests that ask for all of the tables on the system. The default and preferred performance setting 'Default Library List' will cause catalog requests to use only the libraries specified in the default library list when going after library information. Setting the LibraryView value to

‘All libraries on the system’ will cause all libraries on the system to be used for catalog requests and may cause significant degradation in response times due to the potential volume of libraries to process.

**References for ODBC**

- *DB2 Universal Database for System i SQL Call Level Interface (ODBC)*
  is found under the System i Information Center under Printable PDFs and Manuals

- The System i Information Center
  Http://publib.boulder.ibm.com/iseries/

- Microsoft ODBC webpage
  http://msdn2.microsoft.com/en-us/library/ms710252.aspx

# Chapter 11. Domino on i

This chapter includes performance information for Lotus Domino on the IBM i operating system. Some of the information previously included in this section has been removed. Earlier versions of the document can be accessed at http://www.ibm.com/systems/i/solutions/perfmgmt/resource.html

**April 2008 Update**:
- Workload Estimator 2008.2

**January 2008 Updates**:
- V6R1
- Domino 8 white papers
- Workload Estimator 2008.1

**V6R1**
V6R1 may provide improvements in processing capability for Domino environments. V6R1 also requires object conversion for all program objects, including the Domino program objects. This conversion occurs when starting a Domino server for the first time after installing V6R1, or after installing Domino on a V6R1 system, and may take a significant amount of time to complete. For more information on Domino support for V6R1, see:
http://www.ibm.com/systems/i/software/domino/support/v6r1.html.

**POWER6 hardware**
Hardware models based on POWER6 processors may provide improvements in processing capability for Domino environments. For systems that use POWER5 and earlier processors, MCU (Mail and Calendar Users) ratings, rather than CPW ratings, are used to compare Domino performance across hardware models. With the introduction of the POWER6 models, it is less necessary to provide separate MCU and CPW ratings. Appendix C provides projected MCU ratings for POWER6 models that were available as of July 2007, but will not provide ratings for newer hardware models. The IBM Systems Workload Estimator should be used for sizing Domino mail and application workloads. When sizing Domino on i, the latest maintenance release of the selected version is assumed.

**Workload Estimator 2008.2**
Domino sizing support has been changed as follows:
- Support for IBM Power System models has been added.
- Domino 8 disk drive projections have been updated.

**Workload Estimator 2008.1**
Domino sizing support has been changed as follows:
- Sametime support has been updated.
- Quickr for Domino support has been added.

The remainder of this chapter provides performance information for Domino environments.

**Additional Resources**
Additional performance information for Domino on i can be found in the following articles, redbooks and redpapers:

- IBM Lotus Notes V8 workloads: Taking performance to a new level, September 2007
  http://www.ibm.com/developerworks/lotus/library/notes8-workloads/index.html

- IBM Lotus Domino V8 server with the IBM Lotus Notes V8 client: Performance, October 2007
  http://www.ibm.com/developerworks/lotus/library/domino8-performance/index.html

- Lotus Domino 7 Server Performance, Part 2, November 2005
  http://www.ibm.com/developerworks/lotus/library/domino7-internet-performance/index.html

- Lotus Domino 7 Server Performance, Part 3, November 2005
  http://www.ibm.com/developerworks/lotus/library/domino7-enterprise-performance/

- Best Practices for Large Lotus Notes Mail Files, October 2005
  http://www.ibm.com/developerworks/lotus/library/notes-mail-files/

- Lotus Domino 7 Server Performance, Part 1, September 2005
  http://www.ibm.com/developerworks/lotus/library/nd7-perform/index.html

- Redbook and Red Paper Resources found at ( http://www.redbooks.ibm.com/ and http://publib-b.boulder.ibm.com/Redbooks.nsf/redpapers/ )
  - Domino 6 for iSeries Best Practices Guide (SG24-6937), March 2004
  - Lotus Domino 6 for iSeries Multi-Versioning Support on iSeries (SG24-6940), March 2004
  - Sizing Large-Scale Domino Workloads on iSeries (redpaper), December 2003
  - Domino 6 for iSeries Implementation (SG24-6592), February 2003
  - Upgrading to Domino 6: The Performance Benefits (redpaper), January 2003
  - Domino for iSeries Sizing and Performance Tuning (SG24-5162), April 2002
  - iNotes Web Access on the IBM eServer iSeries Server (SG24-6553), February 2002

---

## 11.1 Domino Workload Descriptions

The Mail and Calendaring Users workload and the Domino Web Access mail scenarios discussed in this chapter were driven by an automated environment which ran a script similar to the mail workloads from Lotus NotesBench. Lotus NotesBench is a collection of benchmarks, or workloads, for evaluating the performance of Domino servers. The results from the Mail and Calendaring Users and Domino Web Access workloads are not official NotesBench tests. The numbers discussed for these workloads may not be used officially or publicly to compare to NotesBench results published for other Domino server environments.

Official NotesBench audit results for System i are discussed in section *11.14 System i NotesBench Audits and Benchmarks*. Audited NotesBench results can be found at http://www.notesbench.org .

- **Mail and Calendaring Users (MCU)**
  Each user completes the following actions an average of every 15 minutes except where noted:

  - Open mail database which contains documents that are 10Kbytes in size.
  - Open the current view
  - Open 5 documents in the mail file
  - Categorize 2 of the documents
  - Send 1 new mail memos/replies 10Kbytes in size to 3 recipients. (every 90 minutes)
  - Mark several documents for deletion

- ❖ Delete documents marked for deletion
- ❖ Create 1 appointment (every 90 minutes)
- ❖ Schedule 1 meeting invitation (every 90 minutes)
- ❖ Close the view

- **Domino Web Access (formerly known as iNotes Web Access)**
  Each user completes the following actions an average of every 15 minutes except where noted:

  - ❖ Open mail database which contains documents that are 10Kbytes in size.
  - ❖ Open the current view
  - ❖ Open 5 documents in the mail file
  - ❖ Send 1 new mail memos/replies 10Kbytes in size to 3 recipients (every 90 minutes)
  - ❖ Mark one document for deletion
  - ❖ Delete document marked for deletion
  - ❖ Close the view

  The Domino Web Access workload scenario is similar to the Mail and Calendaring workload except that the Domino mail files are accessed through HTTP from a Web browser and there is no scheduling or calendaring taking place. When accessing mail through Notes, the Notes client performs the majority of the work. When a web browser accesses mail from a Domino server, the Domino server bears the majority of the processing load. The browser's main purpose is to display information.

## 11.2 Domino 8

Domino 8 may provide performance improvements for Notes clients. Test results comparing Domino performance with Domino 7 and Domino 8 have been published in a 2-part series of articles. The following links refer to these articles:

- IBM Lotus Notes V8 workloads: Taking performance to a new level, September 2007
  http://www.ibm.com/developerworks/lotus/library/notes8-workloads/index.html

- IBM Lotus Domino V8 server with the IBM Lotus Notes V8 client: Performance, October 2007
  http://www.ibm.com/developerworks/lotus/library/domino8-performance/index.html

The most up-to-date sizing information on Domino 8 can be found in the Workload Estimator.

## 11.3 Domino 7

Domino 7 provides performance improvements for both Notes and Domino Web Access clients. Test results comparing Domino performance with Domino 6.5 and Domino 7 have been published in a 3-part series of articles titled *Domino 7 Server Performance*. The results show that Domino 7 reduces the amount of CPU required for a given number of users and workload rate as compared with Domino 6.5. The articles also show that the added function in the new Domino 7 mail templates do require some extra processing resources. Results with Domino 7 using the Domino 7 templates show improvements over Domino 6.5 with the Domino 6 mail templates, while Domino 7 with the Domino 6 template provides the

optimal performance but of course without the function provided in the Domino 7 templates. The following links refer to these articles:

- Lotus Domino 7 Server Performance, Part 1, September 2005
  http://www.ibm.com/developerworks/lotus/library/nd7-perform/index.html

- Lotus Domino 7 Server Performance, Part 2, November 2005
  http://www.ibm.com/developerworks/lotus/library/domino7-internet-performance/index.html

- Lotus Domino 7 Server Performance, Part 3, November 2005
  http://www.ibm.com/developerworks/lotus/library/domino7-enterprise-performance/

Additional improvements have been made in Domino 7 to support more users per Domino partition. Internal benchmark tests have been run with as many as 18,000 Notes clients in a single partition. While we understand most customers will not configure a single Domino server to regularly run that number of clients, improvements have been made to enable this type scaling within a Domino partition if desired. A recently published audit report (January 2006) for the System i5 595 demonstrates that 250,500 R6Mail users were run using only 14 Domino mail partitions with most running 18,000 users each.

**Domino Domain Monitor**

Included with Domino 7 is the Domino Domain Monitoring facility which provides a means to monitor and determine the health of an entire domain at a single location and quickly resolve problems. Some of the System i guidelines included for acceptable faulting rates have shown to be a bit aggressive, such that customers have reported that memory alerts and alarms are being triggered even though system performance and response times are acceptable. Work is in progress to make adjustments to future versions of the tool to adjust the faulting guidelines. If you are running this tool and experience alerts for high faulting rates (above 100 per processor), the alerts can be disregarded if you are experiencing acceptable response time and system performance.

## 11.4 Domino 6

Domino 6 provided some very impressive performance improvements over Domino 5, both for workloads we've tested in our lab and for customers who have already deployed Domino 6 on iSeries. In this section we'll provide data showing these improvements based on testing done with the Mail and Calendaring User and Domino Web Access workloads.

**Notes client improvements with Domino 6**
Using the Mail and Calendaring User workload, we compared performance using Domino 5.0.11 and Domino 6. The table below summarizes our results.

| Domino Version | Number of Domino Web Access users | Average CPU Utilization | Average Response Time | Average Disk Utilization |
|---|---|---|---|---|
| | | | | |
| Domino 5.0.11 | 2,000 | 41.5% | 96ms | <1% |
| Domino 6 | 2,000 | 24.0% | 64ms | <1% |
| | | | | |
| Domino 5.0.11 | 3,800 | 19.4% | 119ms | <1% |
| Domino 6 | 3,800 | 11.0% | 65ms | <1% |
| | | | | |
| Domino 5.0.11 | 20,000 | 96.2% | >5sec | <1% |
| Domino 6 | 20,000 | 51.5% | 72ms | <1% |

The 3000 user comparison above was done on an iSeries model i270-2253 which has a 2-way 450MHz processor. This system was configured with 8 Gigabytes (GB) of memory and 12 18GB disk drives configured with RAID5. Notice the 30% improvement in CPU utilization with Domino 6, along with a substantial improvement in response time.

The 8000 user comparison was done on a model i810-2469 which has a 2-way 750MHz processor. The system had 24 8.5GB disk drives configured with RAID5. In this test we notice a slightly greater than 30% improvement in CPU utilization as well as a significant reduction in response time with Domino 6. For this comparison we intentionally created a slightly constrained main storage (memory) environment with 8GB of memory available for the 8000 users. We found that we needed to add 13% more memory, an additional 1GB in this case, when running with Domino 6 in order to achieve the same paging rates, faulting rates, and average disk utilization as the Domino 5.0.11 test. In Domino 6 new memory caching techniques are being used for the Notes client to improve response time and may require additional memory.

Both comparisons shown in the table above were made using single Domino partitions. Similar improvements can be expected for environments using multiple Domino partitions.

**Domino Web Access client improvements with Domino 6**

Using the Domino Web Access workload, we compared performance using Domino 5.0.11 and Domino 6. The table below summarizes our results.

| Domino Version | Number of Mail and Calendaring Users | Average CPU Utilization | Average Response Time | Average Disk Utilization |
|---|---|---|---|---|
| | | | | |
| Domino 5.0.11 | 3,000 | 39.4% | 26ms | 7.1% |
| Domino 6 | 3,000 | 27.6% | 18ms | 5.2% |
| | | | | |
| Domino 5.0.11 | 8,000 | 69.7% | 67ms | 25.2% |
| Domino 6* | 8,000 | 46.7% | 46ms | 26.1% |

* Additional memory was added for this test

Notice that Domino 6 provides at least a 40% CPU improvement in each of the Domino Web Access comparisons shown above, along with significant response time reductions. The comparisons shown above were made on systems with abundant main storage and disk resources so that CPU was the only constraining factor. As a result, the average disk utilization during all of these tests was less than one percent. The purpose of the tests was to compare iNotes Web Access performance using Domino 5.0.11 and Domino 6.

The 2000 user comparison was done on a model i825-2473 with 6 1.1GHz POWER4 processors, 45GB of memory, and 60 18GB disk drives configured with RAID5, in a single Domino partition. The 3800 user comparison used a single Domino partition on a model i890-0198 with 32 1.3GHz POWER4 processors. This system had 64GB of memory and 89 18GB disk drives configured with RAID5 protection. The 20,000 user comparison used ten Domino partitions, also on an i890-0198 32-way system with 1.3GHz POWER4 processors. This particular system was equipped with 192GB of memory and 360 18GB disk drives running with RAID5 protection.

In addition to the test results shown above, many more measurements were performed to study the performance characteristics of Domino 6. One form of tests conducted are what we call "paging curves." To accomplish the paging curves, a steady state was achieved using the workload. Then, over a course of several hours, we gradually reduced the main storage available to the Domino server(s) and observed the effect on paging rates, faulting rates, and response times. These tests allowed us to build a performance curve of the amount of memory available per user versus the paging rate and response time. Based on a paging curve study of the Domino Web Access workload on Domino 6, we determined that, similar to the Mail and Calendaring Users workload, some additional memory was required in order to achieve the same faulting and paging rates as with Domino 5.0.11.

## 11.5 Response Time and Megahertz relationship

The iSeries models and processor speeds described in this section are obviously dated, but the concepts and relationships of response time and megahertz (and gigahertz) described herein are still applicable.

**NOTE**: When comparing models which have different processors types, such as SSTAR, POWER4 and POWER5 it is important to use appropriate rating metrics (see Appendix C) or a sizing tool such as the IBM Systems Workload Estimator. The POWER4 and POWER5 processors have been designed to run at significantly higher MHz than SSTAR processors, and the MHz on SSTAR does not compare directly to the MHz on POWER4 or POWER5.

In general, Domino-related processing can be described as compute intensive (See Appendix C for more discussion of compute intensive workloads). That is, faster processors will generally provide lower response times for Domino processing. Of course other factors besides CPU time need to be considered when evaluating overall performance and response time, but for the CPU portion of the response time the following applies: faster megahertz processors will deliver better response times than an "equivalent" total amount of megahertz which is the sum of slower processors. For example, the 270-2423 processor is rated at 450MHz and the 170-2409 has 2 processors rated at 255MHz; the 1-way 450MHz processor will provide better response time than a 2-way 255MHz processor configuration. The 540MHz, 600MHz, and 750MHz processors perform even faster. Figure 11.3 below depicts the response time performance for three processor types over a range of utilizations. Actual results will vary based on the type of workload being performed on the system.

Using a web shopping application, we measured the following results in the lab. In tests involving 100 web shopping users, the 2-way 170-2409 ran at 71.5% CPU utilization with 0.78 seconds average response time. The 1-way 450MHz 270-2423 ran at 73.6% CPU with average response time of 0.63 seconds. This shows a response time improvement of approximately 20% near 70% CPU utilization which corresponds with the data shown in Figure 11.3. Response times at lower CPU utilizations will see even more improvement from faster processors. The 270-2454 was not measured with the web

shopping application, but would provide even better response times than the 270-2423 as projected in Figure 11.3.

When using MHz alone to compare performance capabilities between models, it is necessary for those models to have the same processor technology and configuration. Factors such as L2 cache and type and speed of memory controllers also influence performance behavior and must be considered. For this reason we recommend using the tables in Appendix C when comparing performance capabilities between iSeries servers. The data in the Appendix C tables take the many performance and processor factors into account and provides comparisons between the iSeries models using three different metrics, CPW, CIW and MCU.

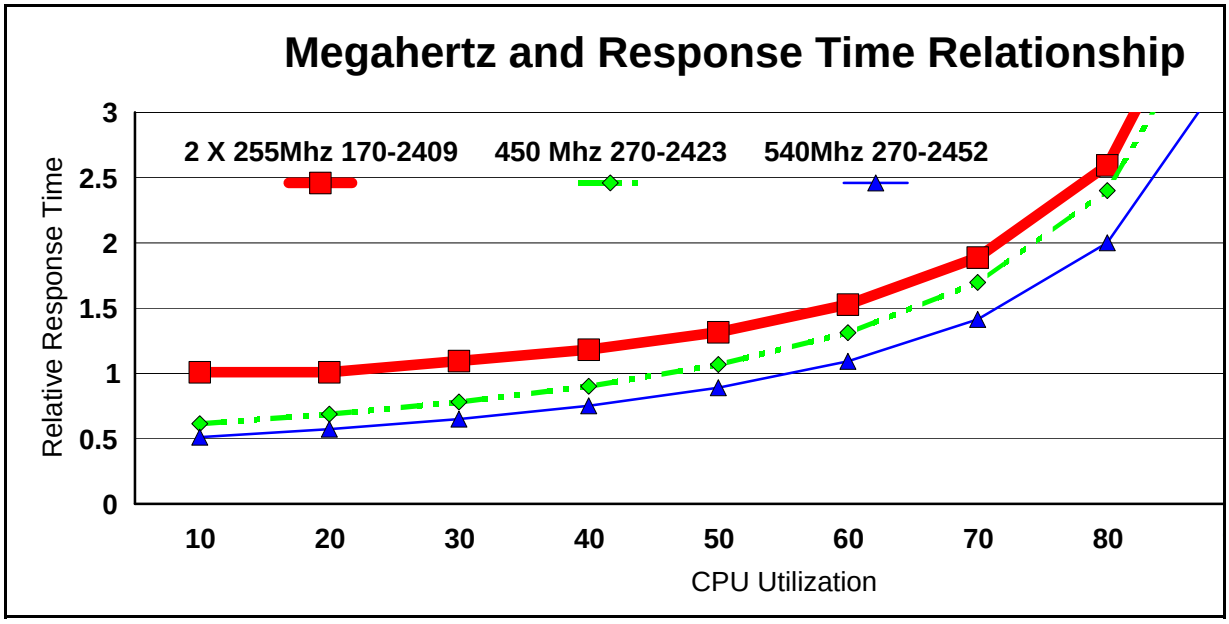



*Figure 11.3 Response Time and Megahertz relationship*

## 11.6 Collaboration Edition and Domino Edition offerings

### Collaboration Edition

The System i Collaboration Edition, announced May 9, 2006, delivers a lower-priced business system to help support the transformation for small and medium sized clients. This edition helps support flexible deployment of **Domino, Workplace, and Portal solutions**, enabling clients to build an on demand computing environment. It provides support for collaboration applications while offering the flexibility of a customizable package of hardware, software, and middleware. Please visit the following site(s) for the additional information:

- http://www.ibm.com/systems/i/hardware/520collaboration/
- http://www.ibm.com/systems/i/solutions/collaboration/

### Domino Edition

The eServer i5 Domino Edition builds on the tradition of the DSD (Dedicated Server for Domino) and the iSeries for Domino offering - providing great price/performance for Lotus software on System i5 and i5/OS. Please visit the following sites for the latest information on Domino Edition solutions:

- http://www.ibm.com/servers/eserver/iseries/domino/
- http://www.ibm.com/servers/eserver/iseries/domino/edition.html

## 11.7 Performance Tips / Techniques

1. Refer to the redbooks listed at the beginning of this chapter which provide Tips and Techniques for tuning and analyzing Domino environments on System i servers.

2. Our mail tests show approximately a 10% reduction in CPU utilization with the system value QPRCMLTTSK(Processor multi-tasking) set to 1 for the pre-POWER4 models. This allows the system to have two sets of task data ready to run for each physical processor. When one of the tasks has a cache miss, the processor can switch to the second task while the cache miss for the first task is serviced. With QPRCMLTTSK set to 0, the processor is essentially idle during a cache miss. This parameter does not apply to the POWER4-based i825, i870, and i890 servers. NOTE: It is recommended to always set QPRCMLTTSK to "1" for the POWER5 models for Domino processing as it has an even greater CPU impact than the 10% described above.

3. It has been shown in customer settings that maintaining a machine pool faulting rate of less than 5 faults per second is optimal for response time performance.

4. iSeries notes.ini / server document settings:
   - Mail.box setting
     Setting the number of mail boxes to more than 1 may reduce contention and reduce the CPU utilization. Setting this to 2, 3, or 4 should be sufficient for most environments. This is in the Server Configuration document for R5.
   - Mail Delivery and Transfer Threads
     You can configure the following in the Server Configuration document:
     - Maximum delivery threads. These pull mail out of mail.box and place it in the users mail file. These threads tended to use more resources than the transfer threads, so we needed to configure twice as many of these so they would keep up.
     - Maximum Transfer threads. These move mail from one server's mail.box to another server's mail.box. In the peer-to-peer topology, at least 3 were needed. In the hub and spoke topology, only 1 was needed in each spoke since mail was transferred to only one location (the hub). Twenty-five were configured for the hubs (one for each spoke).
     - Maximum concurrent transfer threads. This is the number of transfer threads from server 'A' to server 'B'. We set this to 1, which was sufficient in all our testing.

   - NSF_Buffer_Pool_Size_MB
     This controls the size of the memory section used for buffering I/Os to and from disk storage. If you make this too small and more storage is needed, Domino will begin using its own memory management code which adds unnecessary overhead since OS/400 already is managing the virtual storage. If it is made too large, Domino will use the space inefficiently and will overrun the main storage pool and cause high faulting. The general rule of thumb is

that the larger the buffer pool size, the higher the fault rate, but the lower the cpu cost. If the faulting rate looks high, decrease the buffer pool size. If the faulting rate is low but your cpu utilization is high, try increasing the buffer pool size. Increasing the buffer pool size allocates larger objects specifically for Domino buffers, thus increasing storage pool contention and making less storage available for the paging/faulting of other objects on the system. To help optimize performance, increase the buffer pool size until it starts to impact the faulting rate then back it down just a little. Changes to the buffer pool size in the Notes.ini file will require the server to be restarted before taking effect. In Domino 8.0 and later releases, the default buffer pool size is 512MB. In earlier releases, if NFS_Buffer_Pool_Size_MB was not set in the notes.ini file, the buffer pool size could be as large as 1.5GB. A buffer pool size that large might cause performance issues.

- Server_Pool_Tasks
  In the NOTES.INI file starting with 5.0.1, you can set the number of server threads in a partition. Our tests showed best results when this was set to 1-2% of the number of active threads. For example, with 3000 active users, the Server_Pool_Tasks was set to 60. Configuring extra threads will increase the thread management cost, and increase your overall cpu utilization up to 5%.

- Route at once
  In the Server Connection document, you can specify the number of normal-priority messages that accumulate before the server routes mail. For our large server runs, we set this to 20. Overall, this decreased the cpu utilization by approximately 10% by allowing the router to deliver more messages when it makes a connection, rather than 1 message per connection.

- Hub-and-spoke topology versus peer-to-peer topology.
  We attempted the large server runs with both a peer-to-peer topology and a hub-and-spoke topology (see the Domino Administrators guide for more details on how to set this up). While the peer-to-peer funtioned well for up to 60,000 users, the hub-and-spoke topology had better performance beyond 60,000 users due to the reduced number of server to server connections (on the order of 50 versus 600) and the associated costs. A hub topology is also easier to manage, and is sometimes necessitated by the LAN or WAN configuration. Also, according to the Domino Administrators guide, the hub-and-spoke topology is more stable.

5. Dedicate servers to a specific task
   This allows you to separate out groups of users. For example, you may want your mail delivered at a different priority than you want database accesses. This will reduce the contention between different types of users. Separate servers for different tasks are also recommended for high availability.

6. MIME format.
   For users accessing mail from both the Internet and Notes, store the messages in both Notes and MIME format. This offers the best performance during mail retrieval because a format conversion is not necessary. NOTE: This will take up extra disk space, so there is a trade-off of increased performance over disk space.

7. Full text indexes
   Consider whether to allow users to create full text indexes for their mail files, and avoid the use of them whenever possible. These indexes are expensive to maintain since they take up CPU processing time and disk space.

8. Replication.
   To improve replication performance, you may need to do the following:
   - Use selective replication
   - Replicate more often so there are fewer updates per replication
   - Schedule replications at off-peak hours
   - Set up replication groups based on replication priority. Set the replication priority to high, medium, or low to replicate databases of different priorities at different times.

9. Unread marks.
   Select "Don't maintain unread marks" in the advanced properties section of Database properties if unread marks are not important. This can save a significant amount of cpu time on certain applications. Depending on the amount of changes being made to the database, not maintaining unread marks can have a significant improvement. Test results in the lab with a Web shopping applications have shown a cpu reduction of up to 20%. For mail, setting this in the NAB decreased the cpu cost by 1-2%. Setting this in all of the user's mail files showed a large memory and cpu reduction (on the order of 5-10% for both). However, unread marks is an often used feature for mail users, and should be disabled only after careful analysis of the tradeoff between the performance gain and loss of usability.

10. Don't overwrite free space
    Select "Don't overwrite free space" in the advanced properties section of Database properties if system security can be maintained through other means, such as the default of PUBLIC *EXCLUDE for mail files. This can save on the order of 1-5% of cpu. Note you can set this for the mail.box files as well.

11. Full vs. Half duplex on Ethernet LAN.
    Ensure the iSeries and the Ethernet switches in the network are configured to enable a full duplex connection in order to achieve maximum performance. Poor performance can result when running half duplex. This seems rather obvious, but the connection may end up running half duplex even if the i5/OS line description is set to full duplex and even if the switch is enabled for full duplex processing. Both the line description duplex parameter and the switch must be set to agree with each other, and typically it is best to use auto-negotiate to achieve this (*AUTO for the duplex parameter in the line description). Just checking the settings is usually not sufficient, a LAN tester must be plugged into the network to verify full vs. half duplex.

12. Transaction Logging.
    Enabling transaction logging typically adds CPU cost and additional I/Os. These CPU and disk costs can be justified if transaction logging is determined to be necessary for server reliability and recovery speed. The redbook listed at the beginning of this chapter, "Domino for iSeries Sizing and Performance Tuning," contains an entire chapter on transaction logging and performance impacts.

## 11.8 Domino Web Access

The following recommendations help optimize your Domino Web Access environment:

1. Refer to the redbooks listed at the beginning of this chapter. The redbook, "iNotes Web Access on the IBM eServer iSeries server," contains performance information on Domino Web Access including the impact of running with SSL.

2. Use the default number of 40 HTTP threads. However, if you find that the *Domino.Threads.Active.Peak* is equal to *Domino.Threads.Total*, HTTP requests may be waiting or the HTTP server to make an active thread idle before handling the request. If this is the case for your environment, increase the number of active threads until *Domino.Threads.Active.Peak* is less than *Domino.Threads.Total*. Remember that if the number of threads is set very large, CPU utilization will increase. Therefore, the number of threads should not exceed the peak by very much.

3. Enable *Run Web Agents Concurrently* on the Internet Protocols HTTP tab in the Server Document.

4. For optimal messaging throughput, enable two MAIL.BOX files. Keep in mind that MAIL.BOX files grow as a messages queue and this can potentially impact disk I/O operations. Therefore, we recommend that you monitor MAIL.BOX statistics such as *Mail.Waiting* and *Mail.Maximum.Deliver.Time*. If either or both statistics increase over time, you should increase the number of active MAIL.BOX files and continue to monitor the statistics.

## 11.9 Domino Subsystem Tuning

The objects needed for making subsystem changes to Domino are located in library QUSRNOTES and have the same name as the subsystem that the Domino servers run in. The objects you can change are:

- Class (timeslice, priority, etc.)
- Subsystem description (pool configuration)
- Job queue (max active)
- Job description

The system supplied defaults for these objects should enable Domino to run with optimal performance. However, if you want to ensure a specific server has better response time than another server, you could configure that server in its own partition and change the priority for that subsystem (change the class), and could also run that server in its own private pool (change the subsystem description).

You can create a class for each task in a Domino server. You would do this if, for example, you wanted mail serving (SERVER task) to run at a higher priority than mail routing (ROUTER task). To enable this level of priority setting, you need to do the following:
1. Create the classes that you want your Domino tasks to use.
2. Modify the following IFS file '/QIBM/USERDATA/LOTUS/NOTES/DOMINO_CLASSES'. In that file, you can associate a class with a task within a given server.
3. Refer to the release notes in READAS4.NSF for details.

## 11.10 Performance Monitoring Statistics

Function to monitor performance statistics was added to Domino Release 5.0.3. Domino will track performance metrics of the operating system and output the results to the server. Type "show stat platform" at the server console to display them. This feature can be enabled by setting the parameter PLATFORM_STATISTICS_ENABLED=1 in the NOTES.INI file and restarting your server and is automatically enabled in some versions of Domino. Informal testing in the lab has shown that the overhead of having statistics collection enabled is quite small and typically not even measurable.

The i5/OS Performance Tools and Collection Services function can be enabled to collect and report Domino performance information by specifying to run the COLSRV400 task in the Domino server notes.ini file parameter: ServerTasks=UPDATE,COLSRV400,ROUTER,COLLECT,HTTP. With V5R4 a new Domino Server Activity section has been added to the Performance Tools Component Report which looks like the following:

```
Component Report                                  10/20/05 16:21:33
                                                  Domino Server Activity                                                  Page   76
Member  . . . : PERFMON01  Model/Serial  . : 595/55-55555       Main storage . . :  374.0 GB  Started . . . . :  10/20/05 07:56:10
  Library . . : NZEN101905 System name . . :MySystemi             Version/Release  :    5/ 4.0  Stopped . . . . :  10/20/05 08:28:00
Partition ID  : 001        Feature Code  . :7487-8966          Int Threshold  . : 100.00 %
Virtual Processors:  16    Processor Units :  16.0
Server : 855626/QNOTES/SERVER
                                    Peak    ------- Mail ------  ------- Database -------  ----- Name lookup ------
 Itv     Tns              CPU      Concur  Pending    Waiting      Cache        Cache        Cache        Cache      URLs
 End    /Hour     Users   Util     Users   Outbound   Inbound      Hits         Lookups      Hits         Lookups    Rcv/Sec
-----  --------  ------  ------  -------  --------  ---------  -----------  -----------  -----------  -----------  -------
07:58   1515420   18001   39.50    18001        23          0            0          264            0            0        0
07:59   1550099   18001   37.25    18001        23          0            0            0            0            0        0
08:00   1536840   18001   31.95    18001        24          0            0            0            0            0        0
08:01   1874520   18001   35.46    18001        24          0            0            0            0            0        0
.
.
.
08:24   1580400   18001   37.21    18001        34          0            0            0            0            0        0
08:25   1589159   18001   34.79    18001        40          0            0            0            0            0        0
08:26   1575959   18001   35.99    18001        40          0            0            0            0            0        0
08:27   1579739   18001   35.31    18001        40          0            0            0            0            0        0
08:28   1516680   18001   36.84    18001        40          0            0            0            0            0        0
          Column                              Average
        ----------------------------  ----------------
        Tns/Hour                         1,531,036
        Users                               18,001
        CPU Util                             36.55
        Peak Concurrent Users               18,001
        Mail Pending Outbound                  829
        Mail Waiting Inbound                     0
        Database Cache Hits                      0
        Database Cache Lookups                 906
        Name Lookup Cache Hits                   0
        Name Lookup Cache Lookups                0

        URLs Rcv/Sec                             0
```

## 11.11 Main Storage Options

V5R3 provides performance improvements for the *DYNAMIC setting for Main Storage Option on stream files. The charts found later in this section show the improved performance characteristics that can be observed with using the *DYNAMIC setting in V5R3.

In V5R2 two new attributes were added to the OS/400 CHGATR command, *DISKSTGOPT and *MAINSTGOPT. In this section we will describe our results testing the *MAINSTGOPT using the Mail and Calendar workload. The allowed values for this attribute include the following:

1. *NORMAL
   The main storage will be allocated normally. That is, as much main storage as possible will be allocated and used. This minimizes the number of disk I/O operations since the information is cached in main storage. If the *MAINSTGOPT attribute has not been specified for an object, this value is the default.

2. *MINIMIZE
   The main storage will be allocated to minimize the space used by the object. That is, as little main storage as possible will be allocated and used. This minimizes main storage usage while increasing the number of disk I/O operations since less information is cached in main storage.

3. *DYNAMIC
   The system will dynamically determine the optimum main storage allocation for the object depending on other system activity and main storage contention. That is, when there is little main storage contention, as much storage as possible will be allocated and used to minimize the number of disk I/O operations. When there is significant main storage contention, less main storage will be allocated and used to minimize the main storage contention. This option only has an effect when the storage pool's paging option is *CALC. When the storage pool's paging option is *FIXED, the behavior is the same as *NORMAL. When the object is accessed through a file server, this option has no effect. Instead, its behavior is the same as *NORMAL.

These values can be used to affect the performance of your Domino environment. As described above, the default setting is *NORMAL which will work similarly to V5R1. However, there is a new default for the block transfer size of stream files which are created in V5R2. Stream files created in V5R2 will use a block transfer size of 16k bytes, versus 32k bytes in V5R1 and earlier. Files created prior to V5R2 will retain the 32k byte block transfer size. To change stream files created prior to V5R2 to use the 16k block transfer size, you can use the CHGATR command and specify the *NORMAL attribute. Testing showed that the 16k block transfer size is advantageous for Domino mail and calendaring function which typically accesses less than 16k at a time. This may affect the performance of applications that access stream files with a random access patterns. This change will likely improve the performance of applications that read and write data in logical I/O sizes smaller than 16k. Conversely, it may slightly degrade the performance of applications that read and write data with a specified data length greater than 16k.

The *MINIMIZE main storage option is intended to minimize the main storage required when reading and writing stream files and changes the block transfer size of the stream file object to 8k. When reading or writing sequentially, main storage pages for the stream file are recycled to minimize the working set size. To offset some of the adverse effects of the smaller block transfer size and the reduce likelihood that a page is resident, *MINIMIZE synchronously reads several pages from disk when a read or write request would cause multiple page faults. Also, *MINIMIZE avoids reading data from disk when the block of data to be written is page aligned and has a length that is a multiple of the page size.

The *DYNAMIC main storage option is intended to provide a compromise between the *NORMAL and *MINIMIZE settings. This option only has an effect when the storage pool is set to *CALC. The Expert Cache feature of the iSeries allows the file system read and write functions to adjust their internal algorithms based on system tuning recommendations. A system with low paging rates will use an algorithm similar to *NORMAL, but when the paging rates are too high due to main storage contention, the algorithm used will be more like *MINIMIZE. When specifying *DYNAMIC, the block transfer size is set to 12k, midway between the value of *NORMAL and *MINIMIZE.

Deciding when it is appropriate to use the CHGATR command to change the *MAINSTGOPT for a Domino environment is not necessarily straightforward. The rest of this section will discuss test results of using the various attributes. For all of the test results shown here for the *MINIMIZE and *DYNAMIC attributes, the CHGATR command was used to change all of the user mail .NSF files being used in the test.

The following is an example of how to issue the command:

CHGATR OBJ( name of object) ATR(*MAINSTGOPT) VALUE(*NORMAL, *MINIMIZE, or *DYNAMIC)

The chart below depicts V5R3-based paging curve measurements performed with the following settings for the mail databases: *NORMAL, *MINIMIZE, and *DYNAMIC.

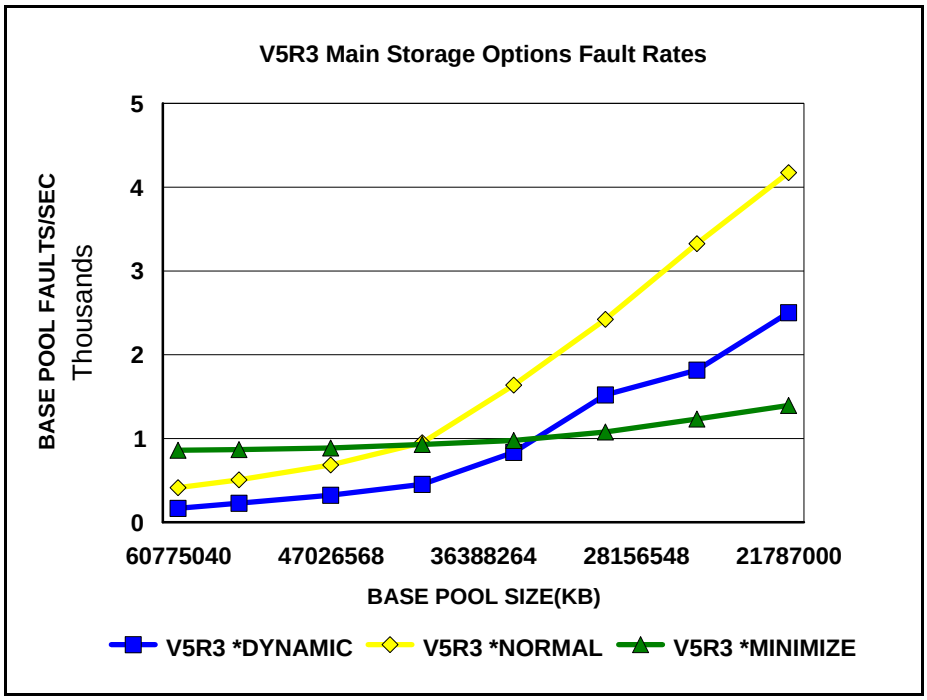



*Figure 11.4 V5R3 Main Storage Options on a Power4 System - Page Fault Rates*

In figures 11.4 and 11.5, results are shown for tests that were performed with a Mail and Calendaring Users workload and various settings for Main Storage Option. The tests started with the users running at steady state with adequate main storage resource available, and then the main storage available to the *base pool containing the Domino servers was gradually reduced The tests used an NSF Buffer Pool Size of 300MB with multiple Domino partitions.

Notice in Figure 11.4 above that as the base pool decreased in size (moving to the right on the chart), the page faulting increased for all settings of main storage option. Using the *DYNAMIC and *NORMAL attributes provided the lowest fault rates when memory was most abundant at the left side of the curve. Moving to the right on the chart as main storage became more constrained, it shows that less page faulting takes place with the *MINIMIZE storage option compared to the other two options. Less page faulting will generally provide better performance.

In V5R3 the performance of *DYNAMIC has been improved and provides a better improvement for faulting rates as compared to *NORMAL than was the case in V5R2. When running with *DYNAMIC in V5R2, information about how the file is being accessed is accumulated for the open instance and adjustments are made for that file based on that data. But when the file is closed and reopened, the algorithm essentially needs to start over. V5R3 includes improvements to keep track of the history of the file access information over open/close instances.

During the tests, the *DYNAMIC and *MINIMIZE settings used up to 5% more CPU resource than *NORMAL.

Figure 11.5 below shows the response time data rather than fault rates for the same test shown in Figure 11.4 for the attributes *NORMAL, *DYNAMIC, and *MINIMIZE.

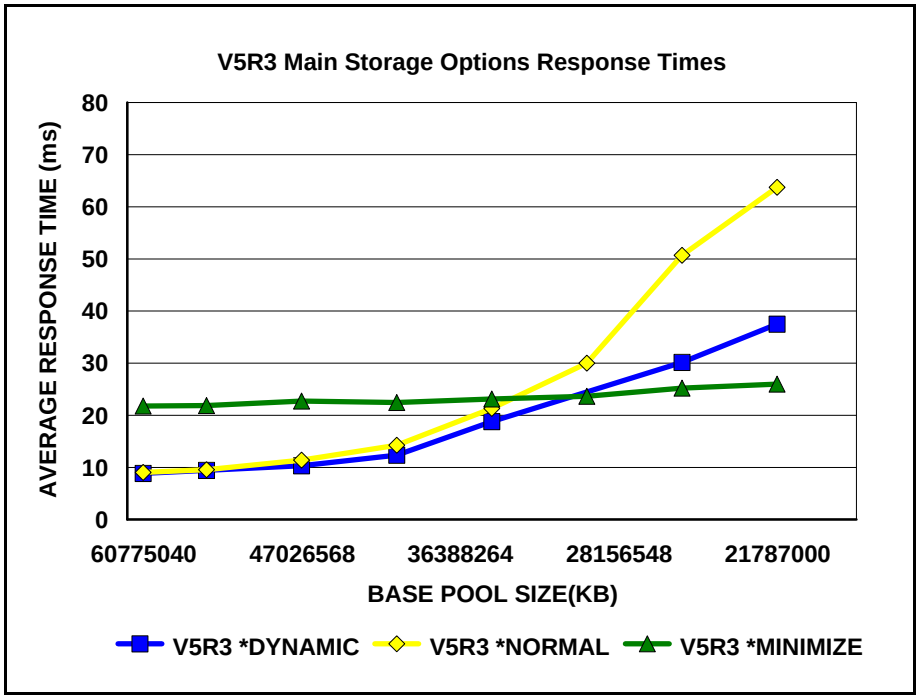



*Figure 11.5 V5R3 Main Storage Options - Response Times*

Notice that there is not an exact correlation between fault rates and response times as shown in Figures 11.4 and 11.5. The *NORMAL and *DYNAMIC option showed the lowest average response times at the left side of the chart where the most main storage was available. As main storage was constrained (moving to the right on the chart), *MINIMIZE provided lower response times.

As is the case with many performance settings, “your mileage will vary” for the use of *DYNAMIC and *MINIMIZE. Depending on the relationship between the CPU, disk and memory resources on a given system, use of the Main Storage Options may yield different results. As has already been mentioned, both *MINIMIZE and *DYNAMIC required up to 5% more CPU resource than *NORMAL. The test environment used to collect the results in Figures 11.4 and 11.5 had an adequate number of disk drives such that disk utilizations were below recommended levels for all tests.

## 11.12 Sizing Domino on System i

To compare Domino processing capabilities for System i servers that use POWER5 and earlier processors, you should use the MCU ratings provided in Appendix C . The ratings are based on the Mail and Calendaring User workload and provide a better means of comparison for Domino processing than do CPW ratings for these earlier models.

*NOTE: MCU ratings should NOT be used directly as a sizing guideline for the number of supported users. MCU ratings provide a relative comparison metric which enables System i models to be compared with each other based on their Domino processing capability. MCU ratings are based on an industry standard workload and the simulated users do not necessarily represent a typical load exerted by "real life" Domino users.*

When comparing models which have different processors types, such as SSTAR, POWER4 and POWER5, it is important to use appropriate rating metrics (see Appendix C) or a sizing tool such as the IBM Systems Workload Estimator. The POWER4 and POWER5 processors have been designed to run at significantly higher MHz than SSTAR processors, and the MHz on SSTAR does not compare directly to the MHz on POWER4 or POWER5.

For sizing Domino mail and application workloads on System i servers, including the new POWER6 models, the recommended method is the IBM Systems Workload Estimator. This tool was previously called the IBM eServer Workload Estimator. You can access the Workload Estimator from the Domino on iSeries home page (select "sizing Information") or at this URL: http://www.ibm.com/eserver/iseries/support/estimator .

The Workload Estimator is typically refreshed 3 to 4 times each year, and enhancements are continually added for Domino workloads. Be sure to read the "What's New" section for updates related to Domino sizing information. The estimator's rich help text describes the enhancements in more detail. Some of the recent additions include: SameTime Application Profiles for Chat, Meeting, and Audio/Video, Domino 6, Transaction Logging, a heavier mail client type, adjustments to take size of database into account, LPAR updates, and enhancement to defining and handling Domino Clustering activity.

Be sure to note the redpaper "Sizing Large-Scale Domino Workloads on iSeries" which is available at: http://www.redbooks.ibm.com/redpapers/pdfs/redp3802.pdf . The paper contains test results for a variety of experiments such as for mail and calendar workloads using different sized documents, and comparisons of the effect of a small versus very large mail database size. A more recent article describes "Best Practices for Large Lotus Notes Mail Files" and is found at: http://www.ibm.com/developerworks/lotus/library/notes-mail-files/

Additional information on sizing Domino HTTP applications for AS/400 can be found at http://www.ibm.com/servers/eserver/iseries/domino/d4appsz.html . Several sizing examples are provided that represent typical Web-enabled applications running on a Domino for AS/400 server. The examples show projected throughput rates for various iSeries servers. To observe transaction rates for a Domino sever you can use the "show stat domino" command and note the Domino.Requests.Per1hour, Domino.Requests.Per1min, and Domino.Requests.Per5min results. The applications described in these examples are included as IBM defined applications in the Workload Estimator.

For more information on performance data collection and sizing, see **Appendix B - iSeries Sizing and Performance Data Collection Tools.**

## 11.13 LPAR and Partial Processor Considerations

Many customers have asked whether the lowest rated i520 models will provide acceptable performance for Domino. Given the CPU intensive nature of most collaborative transactions, the 500 CPW and 600 CPW models may not provide acceptable response times for these types of workloads. The issue is one of response time rather than capacity. So even if the anticipated workload only involves a small number of

users or relatively low transaction rates, response times may be significantly higher for a small LPAR (such as 0.2 processor) or partial processor model as compared to a full processor allocation of the same technology. The IBM Systems Workload Estimator will not recommend the 500 CPW or 600 CPW models for Domino processing.

Be sure to read the section "Accelerator for System i5" in Chapter 6, Web Server and WebSphere Performance. That section describes the new "Accelerator" offerings which provide improved performance characteristics for the i520 models. In particular, note Figure 6.6 to observe potential response time differences for a 500 CPW or 600 CPW model as compared with a higher rated or Accelerated CPW model for a CPU intensive workload.

## 11.14 System i NotesBench Audits and Benchmarks

NotesBench audit reports can be accessed at www.notesbench.org . The results can also be viewed on-line at www.ideasinternational.com/benchmark/bench.html#NotesBench .

# Chapter 12. WebSphere MQ for iSeries

## 12.1 Introduction

The WebSphere MQ for iSeries product allows application programs to communicate with each other using messages and message queuing. The applications can reside either on the same machine or on different machines or platforms that are separated by one or more networks. For example, iSeries applications can communicate with other iSeries applications through WebSphere MQ for iSeries, or they can communicate with applications on other platforms by using WebSphere MQ for iSeries and the appropriate MQ Series product(s) for the other platform (HP-UX, OS/390, etc.).

MQ Series supports all important communications protocols, and shields applications from having to deal with the mechanics of the underlying communications being used. In addition, MQ Series ensures that data is not lost due to failures in the underlying system or network infrastructure. Applications can also deliver messages in a time independent mode, which means that the sending and receiving applications are decoupled so the sender can continue processing without having to wait for acknowledgement that the message has been received.

This chapter will discuss performance testing that has been done for Version 5.3 of WebSphere MQ for iSeries and how you can access the available performance data and reports generated from these tests. A brief list of conclusions and results are provided here, although it is recommended to obtain the reports provided for a more comprehensive look at WebSphere MQ for iSeries performance.

## 12.2 Performance Improvements for WebSphere MQ V5.3 CSD6

WebSphere MQ V5.3 CSD6 introduces substantial performance improvements at queue manager start and during journal maintenance.

**Queue Manager Start Following an Abnormal End**
WebSphere MQ cold starts by customers in the field are a common occurrence after a queue manager ends abnormally because the time needed to clean up outstanding units of work is lengthy (or worse, because the restart does not complete). Note that during a normal shutdown, messages in the outstanding units of work would be cleaned up gracefully.

In tests done in our Rochester development lab, we simulated a large customer environment with 50-500 customers connected, each with an outstanding unit of work in progress, and then ended the queue manager abnormally. These tests showed that with the performance enhancement applied, a queue manager start that previously took hours to complete finished in less than three minutes. Overall, we saw 90% or greater improvement in start times in these cases.

**Checkpoint Following a Journal Receiver Roll-over**
Our goal in this case was to improve responsiveness and throughput with regards to persistent messaging, and reduce the amount of time WebSphere MQ is unavailable during the checkpoint taken after a journal receiver roll-over. Tests were done in the Rochester lab with several different journal receiver sizes and various numbers of journal receivers in the chain in order to assess the impact of this performance enhancement. Our results showed up to a 90% improvement depending on the size and number of journal receivers involved, with scenarios having larger amounts of journal data receiving the most benefit. This

enhancement should allow customers to run with smaller, more manageable, receivers with less concern about the checkpoint taken following a receiver roll-over during business hours.

## 12.3 Test Description and Results

Version 5.3 of WebSphere MQ for iSeries includes several performance enhancements designed to significantly improve queue manager throughput and application response time, as well as improve the overall throughput capacity of MQ Series. Measurements were done in the IBM Rochester laboratory with assistance from IBM Hursley to help show how Version 5.3 compares to Version 5.2 of MQ Series for iSeries.

The workload used for these tests is the standard CSIM workload provided by Hursley to measure performance for all MQ Series platforms. Measurements were done using both client-server and distributed queuing processing. Results of these tests, along with test descriptions, conclusions, recommendations and tips and techniques are available in support pacs at the following URL: http://v06dbl07.hursley.ibm.com/hursley/hiumqweb.nsf/pages/WMQPerformanceTeamHome

From this page, you can select to view all performance support pacs. The most current support pac document at this URL is the “WebSphere MQ for iSeriesV5.3- Performance Evaluations”. This document contains performance highlights for V5.3 of this product, and includes measurement data, performance recommendations, and performance tips and techniques.

## 12.4 Conclusions, Recommendations and Tips

Following are some basic performance conclusions, recommendations and tips/techniques to consider for WebSphere MQ for iSeries. More details are available in the previously mentioned support pacs.

- MQ V5.3 shows an improvement in peak throughput over MQ V5.2 for persistent and nonpersistent messaging, both in client-server and distributed messaging environments. The peak throughput for persistent messaging improved by 15-20%, while for nonpersistent messaging, the peak increased by about 5-10%.

- Tests were also done to determine how many driving applications could be run with a reduced rate of messages per second. The purpose of these tests was not to measure peak throughput, but instead how many of these applications could be running and still achieve response times under 1 second. Compared to MQ Series V5.2, WebSphere MQ for iSeries V5.3 shows an improvement of 40-70% in the number of client-server applications that can be driven in this manner, and an improvement of about 10% in the number of distributed applications.

- Use of a trusted listener process generally results in a reduction in CPU utilization of 5-10% versus using the standard default listener. In addition, the use of trusted applications can result in reductions in CPU of 15-40%. However, there are other considerations to take into account prior to using a trusted listener or applications. Refer to the “Other Sources of Information” section below to find other references on this subject.

- MQ performance can be sensitive to the amount of memory that is available for use by this product. If you are seeing a significant amount of faulting and paging occurring in the memory pools where

applications using MQ Series are running, you may need to consider adding memory to these pools to help performance.

- Nonpersistent messages use significantly less CPU and IO resource than persistent messages do because persistent messages use native journaling support on the iSeries to ensure that messages are recoverable. Because of this, persistent messages should not be used where nonpersistent messages will be sufficient.

- If persistent messages are needed, the user can manually create the journal receiver used by MQ Series on a user ASP in order to ensure best overall performance (MQ defaults to creating the receiver on the system ASP). In addition, the disk arms and IOPs in the user ASP should have good response times to ensure that you achieve maximum capacities for your applications that use persistent messages.

## Other Sources of Information

In addition to the above mentioned support pacs, you can refer to the following URL for reference guides, online manuals, articles, white papers and other sources of information on MQ Series:
http://www.ibm.com/software/ts/mqseries/

# Chapter 13. Linux on iSeries Performance

## 13.1 Summary

Linux on iSeries expands the iSeries platform solutions portfolio by allowing customers and software vendors to port existing Linux applications to the iSeries with minimal effort. But, how does it shape up in terms of performance? What does it look like generally and from a performance perspective? How can one best configure an iSeries machine to run Linux?

### Key Ideas

- "Linux is Linux." Broadly speaking, Linux on iSeries has the same tools, function, look-and-feel of any other Linux.
- Linux operates in its own independent partition, though it has some dependency on OS/400 for a few key services like IPL ("booting").
- Virtual LAN and Virtual Disk provide differentiation for iSeries Linux.
- Shared Processors (fractional CPUs) provides additional differentiation.
- Linux on iSeries provides a mechanism to port many UNIX and Linux applications to iSeries.
- Linux on iSeries particularly permits Linux-based middleware to exploit OS/400 function and data in a single hardware package.
- Linux on iSeries is available on selected iSeries hardware (see IBM web site for details).
- Linux is not dependent *per se* on OS/400 releases. Technically, any Linux distribution could be hosted by any of the present two releases (V5R1 or V5R2) that allow Linux. It becomes a question of service and support. Users should consult product literature to make sure there is support for their desired combination.
- Linux and other Open Source tools are almost all constructed from a single Open Source compiler known as gcc. Therefore, the quality of its code generation is of significant interest. Java is a significant exception to this, having its own code generation.

## 13.2 Basic Requirements -- Where Linux Runs

For various technical reasons, Linux may only be deployed on systems with certain hardware facilities.

These are:

- **Logical partitioning (LPAR).** Linux is not part of OS/400. It needs to have its own partition of the system resources, segregated from OS/400 and, for that matter, any other Linux partitions. A special software feature called the Hypervisor keeps each partition operating separately.
- **"Guest" Operating System Capability**. This begins in V5R1. Part of the iSeries Linux freedom story is to run Linux as Linux, including code from third parties running with root authority and other privilege modes. By definition, such code is not provided by IBM. Therefore, to keep OS/400 and Linux segregated from each other, a few key hardware facilities are needed that are not present on earlier models. (When all partitions run OS/400, the hypervisor's task is simplified, permitting older iSeries and AS/400 to run LPAR).

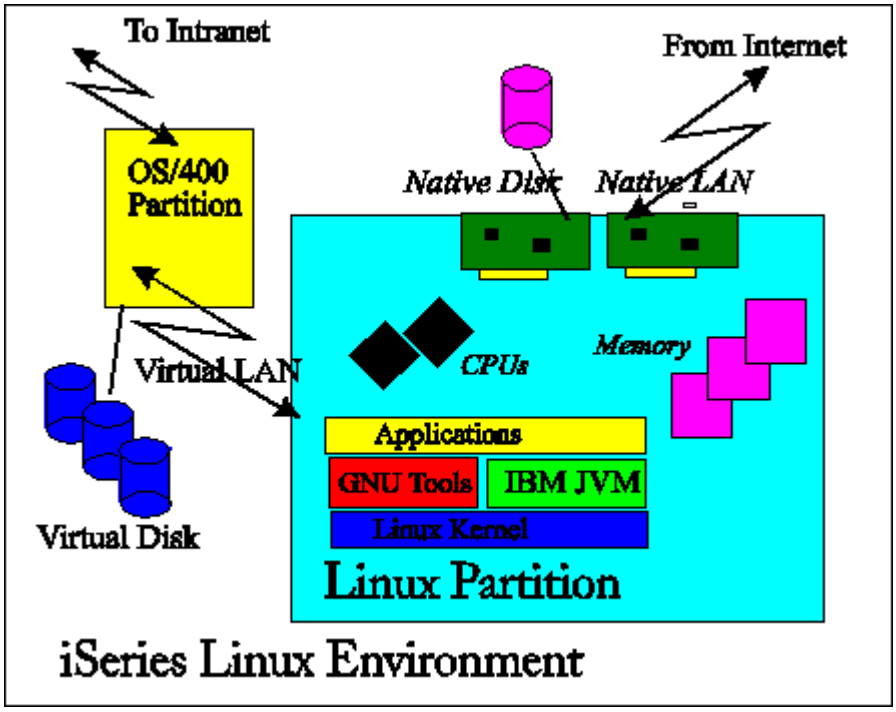
In addition, some models and processor feature codes can run Linux more flexibly than others. The two key features that not all Linux-capable processors support are:

- **Shared Processors.** This variation of LPAR allows the Hypervisor to use a given processor in multiple partitions. Thus, a uni-processor might be divided in various fractions between (say) three LPAR partitions. A four way SMP might give 3.9 CPUs to one partition and 0.1 CPUs to another. This is a large and potentially profitable subject, suitable for its own future paper. Imagine consolidating racks of old, under utilized servers to several partitions, each with a fraction of an iSeries CPU driving it.
- **Hardware Multi-tasking**. This is controlled by the system-wide value QPRCMLTTSK, which, in turn, is controlled by the primary partition. Recent AS/400 and iSeries machines have a feature called hardware multi-tasking. This enables OS/400 (or, now, Linux) to load two jobs (tasks, threads, Linux processes, etc.) into the CPU. The CPU itself will then alternate execution between the two tasks if one task waits on a hardware resource (such as during a cache miss). Due to particular details of some models, Linux cannot run with this enabled. If so, as a practical matter, the entire machine must run with it disabled. In machines where Linux supports this, the choice would be based on experience -- enabling hardware multi-tasking usually boosts throughput, but on occasion would be turned off.

Which models and feature codes support Linux at all and which enable the specific features such as shared processors and hardware multi-tasking are revealed on the IBM iSeries Linux web site.

---

## 13.3 Linux on iSeries Technical Overview




**Linux on iSeries Architecture**

iSeries Linux is a program-execution environment on the iSeries system that provides a traditional memory model (not single-level store) and allows direct access to machine instructions (without the mapping of MI architecture). Because they run in their own partition on a Linux Operating System, programs running in iSeries Linux *do* have direct access to the full capabilities of the user-state and even most supervisor state architecture of the original PowerPC architecture. They do *not* have access to the single level store and OS/400 facilities. To reach OS/400 facilities requires some sort of machine-to-machine interface, such as sockets. A high speed Virtual LAN is available to expedite and simplify this communication.

Storage for Linux comes from two sources: Native and Virtual disks (the latter implemented as OS/400 Network Storage). Native access is provided by allocating ordinary iSeries hard disk to the Linux partition. Linux can, by a suitable and reasonably conventional mount point strategy, intermix both native and virtual disks. The Virtual Disk is analogous to some of the less common Linux on Intel distributions where a Linux file system is emulated out of a large DOS/Windows file, except that on OS/400, the storage is automatically "striped" to multiple disks and, ordinarily, RAIDed.

Linux partitions can also have virtual or native local area networks. Typically, a native LAN would be used for communications to the outside world (including the next fire wall) and the virtual LAN would be used to communicate with OS/400. In a full-blown DMZ ("demilitarized zone") solution, one Linux application partition could provide a LAN interface to the outer fire wall. It could then talk to a second providing the inner fire wall, and then the second Linux partition could use virtual LAN to talk to OS/400 to obtain OS/400 services like data base. This could be done as three total Linux partitions and an OS/400 partition in the back-end.

See "The Value of Virtual LAN and Virtual Disk" for more on the virtual facilities.

**Linux on iSeries Run-time Support**

Linux brings significant support including X-Windows and a large number of shells and utilities. Languages other than C (e.g. Perl, Python, PHP, etc.) are also supported. These have their own history and performance implications, but we can do no more than acknowledge that here. There are a couple of generic issues worth highlighting, however.

Applications running in iSeries Linux work in ASCII. At present, no Linux-based code generator supports EBCDIC nor is that likely. When talking from Linux to OS/400, care must be taken to deal with ASCII/EBCDIC questions. However, for a great fraction of the ordinary Internet and other sockets protocols, it is the OS/400 that is required to shoulder the burden of translation -- the Linux code can and should supply the same ASCII information it would provide in a given protocol. Typically, the translation costs are on the order of five percent of the total CPU costs, usually on the OS/400 side.

iSeries Linux, as a regular Linux distribution, has as much support for Unicode as the application itself provides. Generally, the Linux kernel itself currently has no support for Unicode. This can complicate the question of file names, for instance, but no more or no less than any other Linux environment. Costs for translating to and from Unicode, if present, will also be around five percent, but this will be comparable to other Linux solutions.

## 13.4 Basic Configuration and Performance Questions

Since, by definition, iSeries Linux means at least two independent partitions, questions of configuration and performance get surprisingly complicated, at least in the sense that not everything is on one operating system and whose overall performance is not visible to a single set of tools.

Consider the following environments:

- A machine with a Linux and an OS/400 partition, both running CPU-bound work with little I/O.
- A machine with a Linux and an OS/400 partition, both running work with much I/O or with Linux running much I/O and the OS/400 partition extremely CPU-bound.

The first machine will tend to run as expected. If Linux has 3 of 4 CPUs, it will consume about 0.75 of the machine's CPW rating. In many cases, it will more accurately be observed to consume 0.75 of the CIW rating (processor bound may be better predicted by CIW, absent specific history to the contrary).

The second machine may be less predictable. This is true for regular applications as well, but it could be much more visible here.

Special problems for I/O bound applications:

- The Linux environment is independently operated.
- Virtual disk, generally a good thing, may result in OS/400 and Linux fighting each other for disk access. This is normal if one simply were deploying two traditional applications on an iSeries, but the partitioning may make this more difficult to observe. In fact, one may not be able to attribute the I/O to “anything” running on the OS/400 side, since the various OS/400 performance tools don’t know about any other partition, much less a Linux one. Tasks representing Licensed Internal Code may show more activity, but attributing this to Linux is not straightforward.
- If the OS/400 partition has a 100 per cent busy CPU for long periods of time, the facilities driving the I/O on the OS/400 side (virtual disk, virtual LAN, shared CD ROM) must fight other OS/400 work for the processor. They will get their share and perhaps a bit more, but this can still slow down I/O response time if the '400 partition is extremely busy over a long period of time.

Some solutions:

- In many cases, awareness of this situation may be enough. After all, new applications are deployed in a traditional OS/400 environment all the time. These often fight existing, concurrent applications for the disk and may add "system" level overhead beyond the new jobs alone. In fact, deploying Virtual Disk in a large, existing ASP will normally optimize performance overall, and would be the first choice. Still, problems may be a bit harder to understand if they occur.
- Existing OS/400 guidelines suggest that disk utilization be kept below 42 per cent for non-load source units. That is, controlling disk utilization for both OS/400 and the aggregate Linux Virtual Disks will also control CPU costs. If this can be managed, sharing an ASP should usually work well.
- However, since Linux is in its own partition, and doesn’t support OS/400 notions of subsystem and job control, awareness may not be enough. Alternate solutions include native disk and, usually better, segregating the Linux Virtual Disk (using OS/400 Network Storage objects) into a separate ASP.

## 13.5 General Performance Information and Results

A limited number of performance related tests have been conducted to date, comparing the performance of iSeries Linux to other environments on iSeries and to compare performance to similarly configured (especially CPU MHz) pSeries running the application in an AIX environment.

### Computational Performance -- C-based code

A factor not immediately obvious is that most Linux and Open Source code are constructed with a single compiler, the GNC (gcc or g++) compiler.

In Linux, computational performance is usually dominated by how the gcc/g++ compiler stacks up against commercial alternatives such as xlc (OS/400 PASE) and ILE C/C++ (OS/400). *The leading cause of any CPU performance deficit for Linux (compared to Native OS/400 or OS/400 PASE) is the quality of the gcc compiler's code generation.* This is widely known in the Open Source community and is independent of the CPU architecture.

Generally, for integer-based applications (general commercial):

- OS/400 PASE (xlc) gives the fastest integer performance.
- ILE C/C++ is usually next
- Linux (gcc) is last.

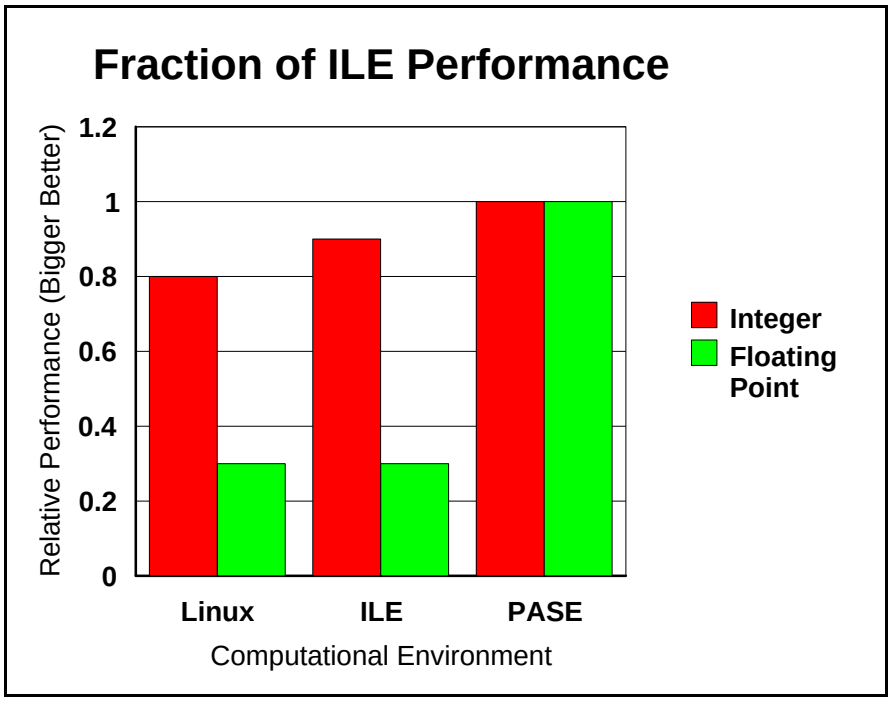
Ordinarily, all would be well within a binary order of magnitude of each other. The difference is close enough that ILE C/C++ sometimes is faster than OS/400 PASE. Linux usually lags slightly more, but is usually not significantly slower.

Generally, for applications dominated by floating point, the rankings change somewhat.

- OS/400 PASE almost always gives the fastest performance.
- Linux and ILE C/C++ often trail substantially. In one measurement, Linux took 2.4 times longer than PASE.

ILE C/C++ floating point performance will be closer to Linux than to OS/400 PASE. Note carefully that most commercial applications *do not* feature floating point.

This chart shows some general expectations that have been confirmed in several workloads.




One virtue of the i870, i890, and i825 machines is that the hardware floating point unit can make up for some of the code generation deficit due to its superior hardware scheduling capabilities.

**Computational Performance -- Java**

Generally, Java computational performance will be dictated by the quality of the JVM used. Gcc performance considerations don't apply (occasional exception: Java Native Methods). Performance on the same hardware with other IBM JVMs will be roughly equal, except that newer JVMs will often arrive a bit later on Linux. The IBM JVM is almost always much faster than the typical open source JVM supplied in many distributions.

**Web Serving Performance**

Work has been done with web serving solutions. Here is some information (primarily useful for sizing, not performance *per se*), which gives some idea of the web serving capacity for static web serving.

| **Number of 840 Processors in Partition** | **0.5** | **1** | **2** | **4** |
|---|---|---|---|---|
| # of web server hits per second, Apache 1.3 | 514 | 1,024 | 1,878 | 3,755 |
| # of web server hits per second, khttpd | 860 | 1,726 | 3,984 | 4,961 |

Here, a model 840 was subdivided into the partition sizes shown and a typical web serving load was used. A "hit" is one web page or one image. The kttpd is a kernel-based daemon available on Linux which serves only static web pages or images. It can be cascaded with ordinary Apache to provide dynamic content as well. The other is a standard Apache 1.3 installation. The 820 or 830 would be a bit less, by about 10 per cent, than the above numbers.

**Network Operations**

Here's some results using Virtual and 100 megabit ethernet. This pattern was repeated in several workloads using 820 and 840 processors:

| TCP/IP Function | 100 megabit Ethernet LAN | Virtual LAN |
|---|---|---|
| Transmit Data | 50-90 megabits per second | 200-400 megabits per second |
| Make Connections | 200-3000 connections per second | 1100-9500 connections per second |

The 825, 870, and 890 should produce slightly higher virtual data rates and nearly the same 100 megabit ethernet rates (since the latter is ultimately limited by hardware). The very high variance in the "make connections" relates, in part, to the fact that several workloads with different complexities were involved.

We also have more limited measurements on Gigabit ethernet showing about 450 megabits per second for some forms of data transmission. For planning purposes, a rough parity between gigabit and Virtual LAN should be assumed.

**Gcc and High Optimization (gcc compiler option -O3)**

The current gcc compiler is used for a great fraction of Linux applications and the Linux kernel. At this writing, the current gcc version is ordinarily 2.95, but this will change over time. This section applies regardless of the gcc version used. Note also that some things that appear to be different compilers (e.g. g++) are front-ends for other languages (e.g. C++) but use gcc for actual generation of code.

Generally speaking, RISC architectures have assumed that the final, production version of an application would be deployed at a high optimization. Therefore, it is important to specify the best optimization level (gcc option -O3) when compiling with gcc or any gcc derivatives. When debugging (-g), optimization is less important and even counterproductive, but for final distribution, optimization often has dramatic performance differences from the base case where optimization isn’t specified.

Programs can run twice as fast or even faster at high versus low optimization. It may be worthwhile to check and adjust Makefiles for performance critical, open source products. Likewise, if compilers other than gcc are deployed, they should be examined to see if their best optimizations are used.

One should check the *man* page (*man gcc* at a command line) to see if other optimizations are warranted. Some potentially useful optimizations are not automatically turned on because not all applications may safely use them.

**The Gcc Compiler, Version 3**

As noted above, many distributions are based on the 2.95 gcc compiler. The more recent 3.2 gcc is also used by some distributions. Results there shows some variability and not much net improvement. To the extent it improves, the gap with ILE should close somewhat. Floating point performance is improved, but proportionately. None of the recommendations in terms of Linux versus other platforms change because the improvement is too inconsistent to alter the rankings, though it bears watching in the future as gcc has more room to improve. This is comparing at -O3 as per the prior section's recommendations.

---

## 13.6 Value of Virtual LAN and Virtual Disk

### Virtual LAN

Virtual LAN is a high speed interconnect mechanism which appears to be an ordinary ethernet LAN as far as Linux is concerned.

There are several benefits from using Virtual LAN:

- *Performance.* It functions approximately on a par with Gigabit ethernet (see previous section, Network Primitives).
- *Cost.* Since it uses built-in processor facilities accessed via the Hypervisor (running on the hosting OS/400 partition), there are no switches, hubs, or wires to deal with. At gigabit speeds, these costs can be significant.
- *Simplification and Consolidation*. It is easy to put multiple Linux partitions on the same Virtual LAN, achieving the same kinds of topologies available in the real world. This makes Virtual LAN ideal for server consolidation scenarios.

The exact performance of Virtual LAN, as is always the case, varies based on items like average IP packet size and so on. However, in typical use, we've observed speeds of 200 to 400 megabits per second on 600 MHz processors. The consumption on the OS/400 side is usually 10 per cent of one CPU or less.

### Virtual Disk

Virtual Disk simulates an arbitrarily sized disk. Most distributions make it "look like" a large, single IDE disk, but that is an illusion. In reality, the disks used to implement it are based on OS/400 Network Storage (*NWSSTG object) and will be allocated from all available (SCSI) disks on the Auxiliary Storage Pool (ASP) containing the Network Storage. By design, OS/400 Single Level Store always "stripes" the data, so Linux files of a nontrivial size are accordingly spread over multiple physical disks. Likewise, a typical ASP on OS/400 will have RAID-5 or mirrored protection, providing all the benefits of these functions without any complexity on the Linux side at all.

Thus, the advantages are:

- *Performance*. Parallel access is possible. Since the data is striped, it is possible for the data to be concurrently read from multiple disks.
- *Reduction in Complexity*. Because it looks like one large disk to Linux, but is typically implemented with RAID-5 and striping, the user does not need to deploy complex strategies such as Linux Volume Management and other schemes to achieve RAID-5 and striping. Moreover, obtaining both strategies (which are, in effect, true by default in OS/400) is more complex still in the Linux environment.

- *Cost.* Because the disk is virtual, it can be created to any size desired. For some kinds of Linux partitions, a single modern physical disk is overkill -- providing far more data than required. These requirements only increase if RAID, in particular, is specified. Here, the Network Storage object can be created to any desired size, which helps keep down the cost of the partition. For instance, for some kinds of middleware function, Linux can be deployed anywhere between 200 MB and 1 GB or so, assuming minimal user data. Physical disks are nowadays much larger than this and, often, much larger than the actual need, even when user/application data is added on.
- *Simplification and Consolidation*. The above advantages strongly support consolidation scenarios. By "right sizing" the required disk, multiple Linux partitions can be deployed on a single iSeries, using only the required amount of disk space, not some disk dictated or RAID-5 dictated minimum. Additional virtual disks can be readily added and they can be saved, copied. etc. using OS/400 facilities.

In terms of performance, the next comparison is compelling, but also limited. Virtual Disk can be much faster than single Native disks. In a really large and complex case, a Native Disk strategy would also have multiple disks, possibly managed by the various Linux facilities available for RAID and striping. Such a usage would be more competitive. But we anticipate that, for many uses of Linux, that level of complexity will be avoided. This makes our comparison fair in the sense that we are comparing what real customers will select between and solutions which, for the iSeries customer, have comparable complexity to deploy.

- 1 disk Intel box, 667 MHz CPU: 5 MB/sec for block writes, 3.4 MB/sec for block reads.
- Virtual Disk, OS/400 1 600 MHz CPU: 112 MB/sec for block writes, 97 MB/sec for block reads

As noted, this is not an absolute comparison. Linux has some file system caching facilities that will moderate the difference in many cases. The absolute numbers are less important than the fact that there is an advantage. The point is: To be sure of this level of performance from the Intel side, more work has to be done, including getting the right hardware, BIOS, and Linux tools in place. Similar work would also have to be done using Native Disk on iSeries Linux. Whereas, the default iSeries Virtual Disk implementation has this kind of capability built-in.

---

## 13.7 DB2 UDB for Linux on iSeries

One exciting development has been the release of DB2 UDB V8.1 for Linux on iSeries. The iSeries now offers customers the choice of an enterprise level database in Linux as well as OS/400.

The choice of which operating environment to use (OS/400 or Linux) will typically be determined by which database a specific application supports. In some cases (e.g., home-grown applications), both operating environments are choices to support the new application. Is performance a reason to select Linux or OS/400 for DB2 UDB workloads?

Initial performance work suggests :

1. If an OLTP application runs well with either of these two data base products, there would not normally be enough performance difference to make the effort of porting from one to the other worthwhile. The OS/400-based DB2 product is a bit faster in our measurements, but not enough to make a compelling difference. Note also that all Linux DB2 performance work to date has used the iSeries virtual storage capabilities where the Linux storage is managed as objects within OS/400. The virtual storage option is

typically recommended because it allows the Linux partitions to leverage the storage subsystem the customer has in the OS/400 hosting partition.

2. As the application gains in complexity, it is probably less likely that the application should switch from one product to the other. Such applications tend to implicitly play to particular design choices of their current product and there is probably not much to gain from moving them between products.

3. As scalability requirements grow beyond a 4-way, the DB2 on OS/400 product provides proven scalability that Linux may not match at this time. If functional requirements of the application require DB2 UDB on Linux and scaling beyond 4 processors, then a partitioned data base and multiple LPARs should be explored.

See also the IBM eServer Workload Estimator for sizing information and further considerations when adding the DB2 UDB for Linux on iSeries workload to your environment.

---

### 13.8 Linux on iSeries and IBM eServer Workload Estimator

At this writing, the Workload Estimator contains the following workloads for Linux on iSeries:

- File Serving
- Web Serving
- Network Infrastructure (Firewall, DNS/DHCP)
- Linux DB2 UDB

These contain estimators for the above popular applications, helpful for estimating system requirements. Consult the latest version of Workload Estimator, including its on-line help text, when specifying a system containing relevant Linux partitions. The workload estimator can be accessed from a web browser at http://www-912.ibm.com/wle/EstimatorServlet.

---

### 13.9 Top Tips for Linux on iSeries Performance

Here's a summary of top tips for improving your Linux on iSeries LPAR performance:

- **Keep up to date on OS/400 PTFs for your hosting partition**. This is a traditional, but still useful recommendation. So far, some substantial performance improvements have been delivered in fixes for Virtual LAN and Virtual Disk in particular.

- **Investigate keeping up to date with your distribution's kernel.** Since these are not offered by IBM, this document cannot make any claims whatever about the value of upgrading the kernel provided by your Linux distributor. That said, it may be worth your while to investigate and see if any kernel updates are provided and whether you, yourself can determine if they aid your performance.

- **If possible, compare your Distribution's versions.** This is a topic well beyond this paper in any detail, but in practice fairly simple. A Linux distributor might offer several versions of Linux at any given moment. Usually, you will wish the latest version, as it should be the fastest. But, if you can

do so, you may wish to compare with the next previous version. This would be especially important if you have one key piece of open source code largely responsible for the performance of a given partition. There is no way of ensuring that a new distribution is actually faster than the predecessor except to test it out. While, formally, no open source product can ever be withdrawn from the marketplace, actual support (from your distributor or possibly other sources) is always a consideration in making such a call.

- **Evaluate upgrading to gcc 3 or sticking with 2.95.** At this writing, the 3.2 version of gcc and perhaps later versions are being delivered, but some other version may be more relevant by the time you read these words. Check with your Linux distributor about when or if they choose to make it available. With sufficiently strong Linux skills, you might evaluate and perform the upgrade to this level yourself for some key applications if it helps them. The distribution may also continue to make 2.95 available (largely for functional reasons). Note also that many distributions will distribute only one compiler. If multiple compilers are shipped with your distribution, and the source isn't dependent on updated standards, you might have the luxury of deciding which to use.

- **Avoid "awkward" LPAR sizes.** If you are running with shared processors, and your sizing recommends one Linux partition to have 0.29 CPUs and the other one 0.65 CPUs, check again. You might be better off running with 0.30 and 0.70 CPUs. The reason this may be beneficial is that your two partitions would tend to get allocated to one processor most of the time, which should give a little better utilization of the cache. Otherwise, you may get some other partition using the processor sometimes and/or your partitions may more frequently migrate to other processors. Similarly, on a very large machine (e.g. an 890), the overall limit of 32 partitions on the one hand and the larger number of processors on the other begins to make shared processors less interesting as a strategy.

- **Use IBM's JVM, not the default Java typically provided**. IBM's PowerPC Java for Linux is now present on most distributions or it might be obtained in various ways from IBM. For both function and performance, the IBM Java should be superior for virtually all uses. On at least one distribution, deselecting the default Java and selecting IBM's Java made IBM's Java the default. In other cases, you might have to set the PATH and CLASSPATH environment variables to put IBM's Java ahead of the one shipped with most distributions.

- **For Web Serving, investigate khttpd.** There is a kernel extension, khttpd, which can be used to serve "static" web pages and still use Apache for the remaining dynamic functionality. Doing so ordinarily improves performance

- **Keep your Linux partitions to a 4-way or less if possible.** There will be applications that can handle larger CPU counts in Linux, and this is improving as new kernels roll out (up to 8-way is now possible). Still, Linux scaling remains inferior to OS/400 overall. In many cases, Linux will run middleware function which can be readily split up and run in multiple partitions.

- **Make sure you have enough storage in the machine pool to run your Virtual Disk function**. Often, an added 512 MB is ample and it can be less. In addition, make sure you have enough CPU to handle your requirements as well. These are often very nominal (often, less than a full CPU for fairly large Linux partition, such as a 4-way), but they do need to be covered. Keep some reserve CPU capacity in the OS/400 partition to avoid being "locked out" of the CPU while Linux waits for Virtual Disk and LAN function.

- **Make sure you have some "headroom" in your OS/400 hosting partition for Virtual I/O.** A rule of thumb would be 0.1 CPUs in the host for every CPU in a Linux partition, presuming it uses a

substantial amount of Virtual I/O. This is probably on the high side, but can be important to have something left over. If the hosting partition uses all its CPU, Virtual I/O may slow substantially.

- **Use Virtual LAN for connections between iSeries partitions whether OS/400 or Linux.** If your OS/400 PTFs are up to date, it performs roughly on a par with gigabit ethernet and has zero hardware cost, no switches and wires, etc.

- **Use Virtual Disk for disk function.** Because virtual disk is spread ("striped") amongst all the disks on an OS/400 ASP, virtual disk will ordinarily be faster. Moreover, with available features like mirroring and RAID-5, the data is also protected much better than on a single disk. Certainly, the equivalent function can be built with Linux, but it is much more complex (especially if both RAID-5 and striping is desired). A virtual disk gives the advantages of both RAID-5 and data "striping" and yet it looks like an ordinary, single hard file to Linux.

- **Use Hardware Multithreading if available.** While this will not always work, Hardware multithreading (a global parameter set for all partitions) will ordinarily improve performance by 10 to 25 per cent. Make sure that it profits all important partitions, not just the current one under study, however. Note that some models cannot run with QPRCMLTTSK set to one ("on") and for the models 825, 870, and 890, it is not applicable.

- **Use Shared Processors, especially to support consolidation**. There is a global cost of about 8 per cent (sometimes less) for using the Shared Processors facility. This is a general Hypervisor overhead. While this overhead is not always visible, it should be planned for as it is a normal and expected result. After paying this penalty, however, you can often consolidate several existing Linux servers with low utilization into a single iSeries box with a suitable partition strategy. Moreover, the Virtual LAN and Virtual Disk provide further performance, functional, and cost leverage to support such uses. Remember that some models do not support Shared Processors.

- **Use spread_lpevents=*n* when using multiple Virtual Processors from a Shared Processor Pool.** This kernel parameter causes processor interrupts for your Linux partition to be spread across *n* processors. Workloads that experience a high number of processor interrupts may benefit when using this parameter. See the Redbooks or manuals for how to set kernel parameters at boot time.

- **Avoid Shared Processors when their benefits are absent**. Especially as larger iSeries boxes are used (larger in terms of CPU count), the benefits of consolidation may often be present without using Shared Processors and its expected overhead penalty. After all, with 16 or more processors, adding or subtracting a processor is now less than 10 per cent of the overall capacity of the box. Similarly, boxes lacking Shared Processor capability may still manage to fit particular consolidation circumstances very well and this should not be overlooked.

- **Watch your "MTU" sizes on LANs.** Normally, they are set up correctly, but it is possible to mismatch the MTU (transmission unit) sizes for OS/400 and Linux whether Virtual or Native LAN. For Virtual LAN, both sides should be 9000. For 100 megabit Native, they should be 1500. These are the values seen in *ifconfig* under Linux. On OS/400, for historical reasons, the correct values are 8996 and 1496 respectively and tend to be called "frame size." If OS/400 says 1496 and Linux says 1500, they are identical. Also, when looking at the OS/400 line description, make sure the "Source Service Access Point" for code AA is also the same as the frame size value. The others aren't critical. While it is certain that the frame sizes on the same device should be identical, it may also be profitable to have all the sizes match. In particular, testing may show that virtual LAN should be changed to 1500/1496 due to end-to-end considerations on critical network paths involving both

Native and Virtual LAN (e.g. from outside the box on Native LAN, through the  partition with the Native LAN, and then moving to a second partition via Virtual LAN then to another).

# Chapter 14. DASD Performance

This chapter discusses DASD subsystems available for the System i platform.

There are two separate considerations. Before IBM i operating system V6R1, one only had to consider particular devices, IOAs, IOPs, and SAN devices. All attached through similar strategies directly to IBM i operating system and were all supported natively.

Starting in IBM iV6R1, however, IBM i operating system will be permitted to become a virtual client of an IBM product known as VIOS. The supported BladeCenter products like the JS12 Express and JS22 Express will only be available in this fashion. For other IBM Power Systems it will be possible to attach all or some of the disks in this manner. This product and its implications will be discussed commencing with section 14.5.

## 14.1 Internal (Native) Attachment.

This section is intended to show relative performance differences in Disk Controllers which I will refer to as IOAs, DASD and IOPs, for customers to compare some of the available hardware. The workload used for our throughput measurements should not be used to gauge the workload capabilities of your system, since it is not a customer like workload.

The workload is designed to concentrate more on DASD, IOAs and IOPs, not the system as a whole. Workload throughput is not a measurement of operations per second but an activity counter in the workload itself. No LPAR's were used, all system resources were dedicated to the testing. The workload is batch and I/O intensive (small block reads and writes).

This chapter refers to disk drives and disk controllers (IOAs) using their CCIN number/code. The CCIN is what the system uses to understand what components are installed and is unique by each device. It is a four character, alphanumeric code. When you use commands in IBM i operating system to print your system configuration like PRTSYSINF or use the WRKHDWRSC *STG command to display hardware configuration information for your storage devices like the 571E or 571F disk controllers you see a listing of CCIN codes.

Note that the feature codes used in IBM's ordering system, e-config tool and inventory records are a four character numeric code which may or may not match the CCIN. IBM will sometimes use different features for the exact same physical device in order to communicate how the hardware is configured to the e-config tool or to provide packaging or pricing structures for the disk drive or IOA. For example, feature code 5738 and 5777 both identify a 571E IOA. A fairly complete list of CCIN and their feature codes can be found in an appendix of the System Builder located near the end of the publication, and a partial list can be found on the following page.

## 14.1.0 Direct Attach (Native)

## 14.1.1 Hardware Characteristics

#### 14.1.1.1 Devices & Controllers

| CCIN Codes | Approximate Size (GB) | RPM | Seek Time (ms) | | Latency (ms) | Max Drive Interface Speed (MB/s) when mounted in a given enclosure | | |
|---|---|---|---|---|---|---|---|---|
| | | | Read | Write | | 5074/5079 | 5094/ 5294 | 5786/ 5787 |
| 6718 | 18 | 10K | 4.9 | 5.9 | 3 | 80 | 80 | NA |
| 6719 | 35 | 10K | 4.7 | 5.3 | 3 | 80 | 160 | NA |
| 4326 | 35 | 15K | 3.6 | 4.0 | 2 | Not Supported | 160 | 320 |
| 4327 | 70 | 15K | 3.6 | 4.0 | 2 | Not Supported | 160 | 320 |
| 4328 | 140 | 15K | 3.6 | 4.0 | 2 | Not Supported | 160 | 320 |
| 4329 | 280 | 15K | 3.6 | 4.0 | 2 | Not Supported | Not Supported | 320 |
| 433B | 70 | 15K | 3.5 | 4.0 | 2 | N/A | N/A | N/A |
| 433C | 140 | 15K | 3.5 | 4.0 | 2 | N/A | N/A | N/A |
| 433D | 280 | 15K | 3.5 | 4.0 | 2 | N/A | N/A | N/A |

| CCIN Codes | (IOA) Feature Codes | Cache non-compressed / up to compressed | Min/Max # of drives in a RAID set | Max Drive Interface Speed supported #1 (MB/s) |
|---|---|---|---|---|
| 5702 | 5705, 5712, 5715, 0624 | NA | NA | 160 |
| 5703 | 5703 | 40 MB | 3/18 | 320 |
| 2757 | 5581, 2757, 5591 | 235 MB / up to 757 | 3/18 | 160 |
| 2780 | 5580, 2780, 5590 | 235 MB write/up to 757<br>256 MB read/up to 1GB | 3/18 | 320 |
| 5709<br>Write cache card for built in IOA | 5709, 5726, 9509 | 16 MB | 3/8 | NA |
| 573D<br>Write cache card for built in IOA | 5727, 5728, 9510 | 40 MB | 3/8 | NA |
| 57B8<br>(Aux cache card 57B7) | 5679 | 175 MB | 3/18 RAID5<br>4/18 RAID6 | 300 |
| 571A | 5736, 5775, 0647 | NA | NA | 320 |
| 571B | 5737, 5776, 0648 | 90 MB | 3/18 RAID5<br>4/18 RAID6 | 320 |
| 571E/574F | 5738, 5777, 5582, 5583 | 390 MB write/up to 1.5GB<br>415 MB read/up to 1.6GB | 3/18 RAID5<br>4/18 RAID6 | 320 |
| 571F/575B | 5739, 5778, 5781, 5782, 5799, 5800 | 390 MB write/up to 1.5GB<br>415 MB read/up to 1.6 GB | 3/18 RAID5<br>4/18 RAID6 | 320 |
| 572C | 572C | NA | NA | 300 |
| 572A | 572A | NA | NA | 300 |

Note: The actual drive interface speed (MB/s) is the minimum value of the maximum supported speeds of the drive, the enclosure and the IOA. Also note that the minimum value for the various drive & enclosure combinations are identified in the above table.

Not all disk enclosures support the maximum number of disks in a RAID set.

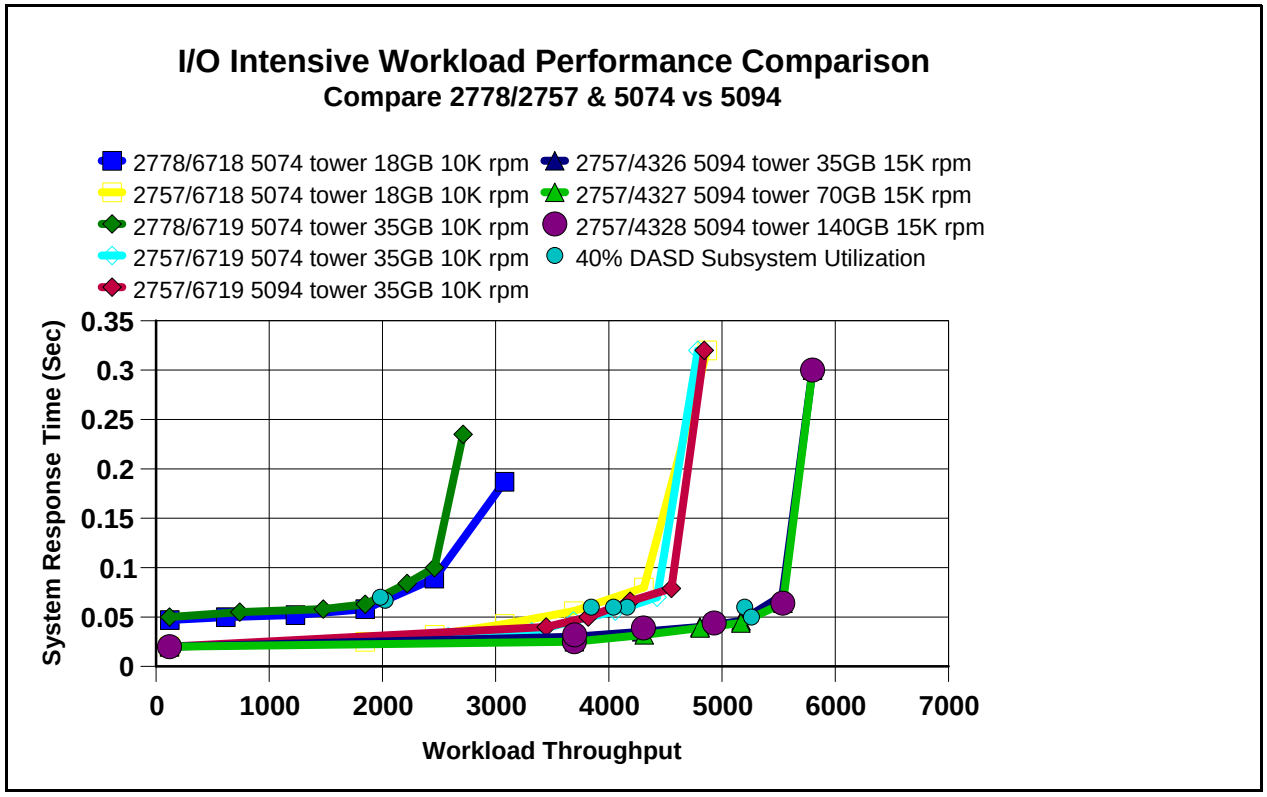
### 14.1.2 iV5R2 Direct Attach DASD

This section discusses the direct attach DASD subsystem performance improvements that were new with the iV5R2 release. These consist of the following new hardware and software offerings :

- 2757 SCSI PCI RAID Disk Unit Controller (IOA)
- 2780 SCSI PCI RAID Disk Unit Controller (IOA)
- 2844 PCI Node I/O Processor (IOP)
- 4326 35 GB 15K RPM DASD
- 4327 70 GB 15K RPM DASD

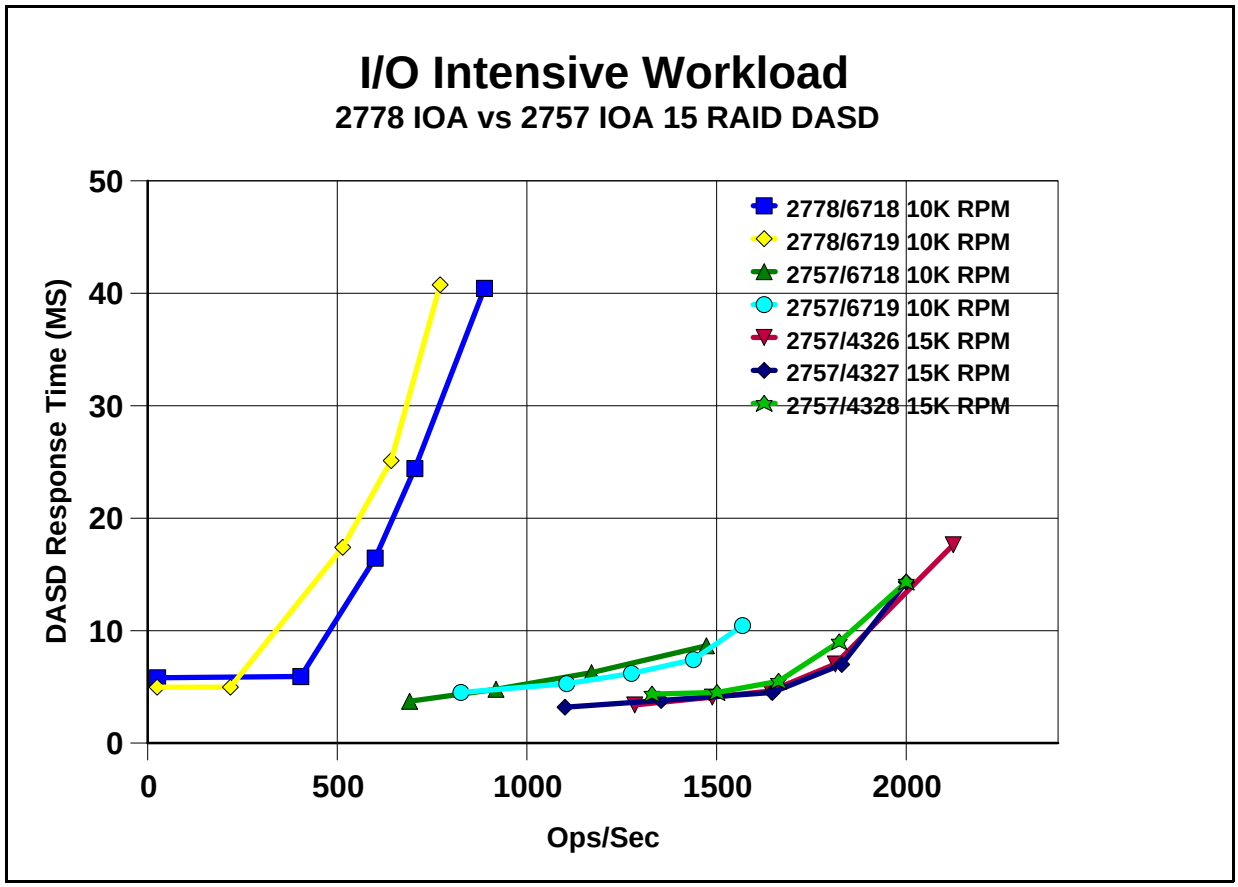
Note: for more information on the older IOAs and IOPs listed here see a previous copy of the Performance Capabilities Reference.

**14.1.2.1**




For our workload we attempt to fill the DASD units to between 40 and 50% full so you are comparing units with more actual data, but trying to keep the relative seek distances similar. The reason is that larger capacity drives can appear to be faster than lower capacity drives in the same environment running the same workload in the same size database. That perceived improvement can disappear, or even reverse depending upon the workload (primarily because of where on the disks the data is physically located).

**14.1.2.2**




| IOA and operation | | Number of 35 GB DASD units (Measurement numbers in GB/HR) | | |
|---|---|---|---|---|
| 2778 IOA | | 15 Units | 30 Units | 45 Units |
| *SAVF | Save | 41 | 83 | 122 |
| | Restore | 41 | 83 | 122 |
| 2757 IOA | | | | |
| *SAVF | Save | 82 | 165 | 250 |
| | Restore | 82 | 165 | 250 |

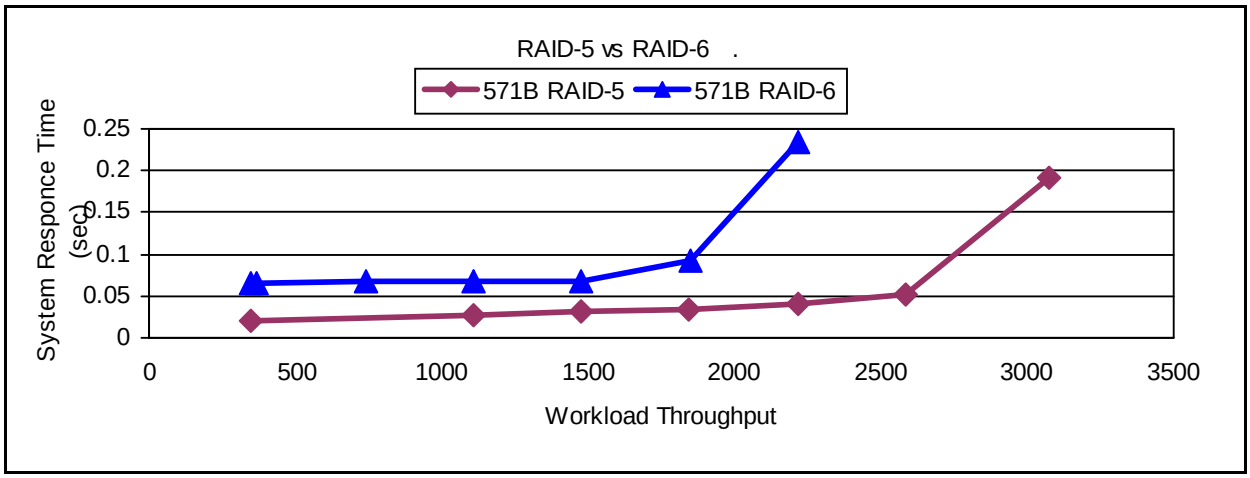
This restrictive test is intended to show the effect of the 2757 IOAs in a backup and recovery environment. The save and restore operations to *SAVF (save files) were done on the same set of DASD, meaning we were reading from and writing to the same 15, 30, and 45 DASD units at the same time. So the number of I/O DASD operations are double when saving to *SAVF. This was not meant to show what can be expected from a backup environment, see chapter 15 for save and restore device information.

### 14.1.3 571B

iV5R4 offers two new options on DASD configuration.

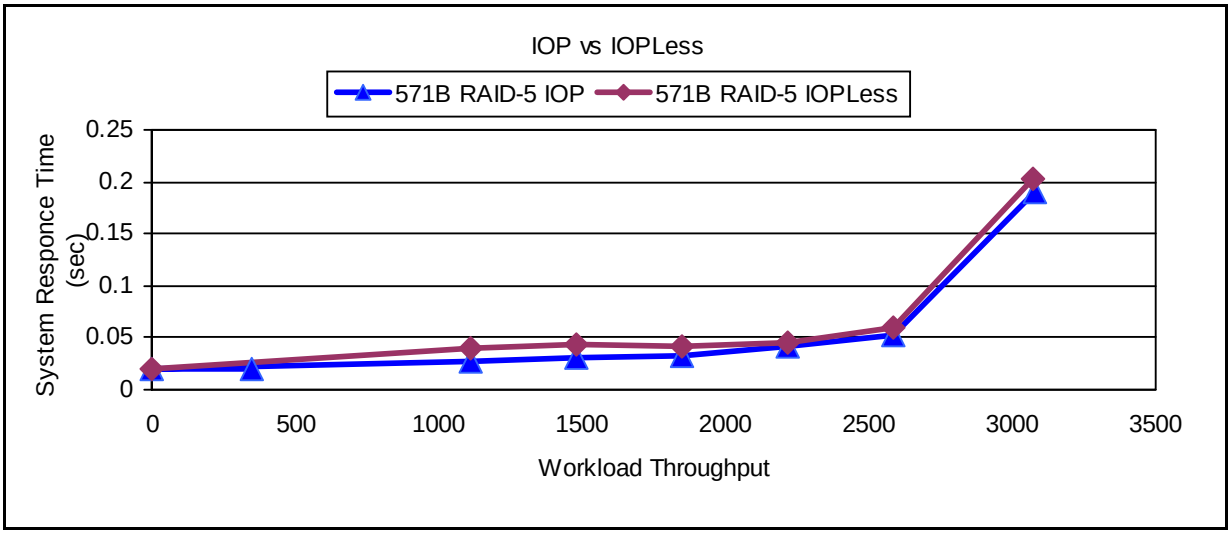
- RAID6 which offers improved system protection on supported IOAs.
- NOTE: RAID6 is supported under iV5R3 but we have chosen to look at performance data on a iV5R4 system.
- IOPLess operation on supported IOAs.

#### 14.1.3.1 571B RAID5 vs RAID6 - 10 15K 35GB DASD




#### 14.1.3.2 571B IOP vs IOPLESS - 10 15K 35GB DASD

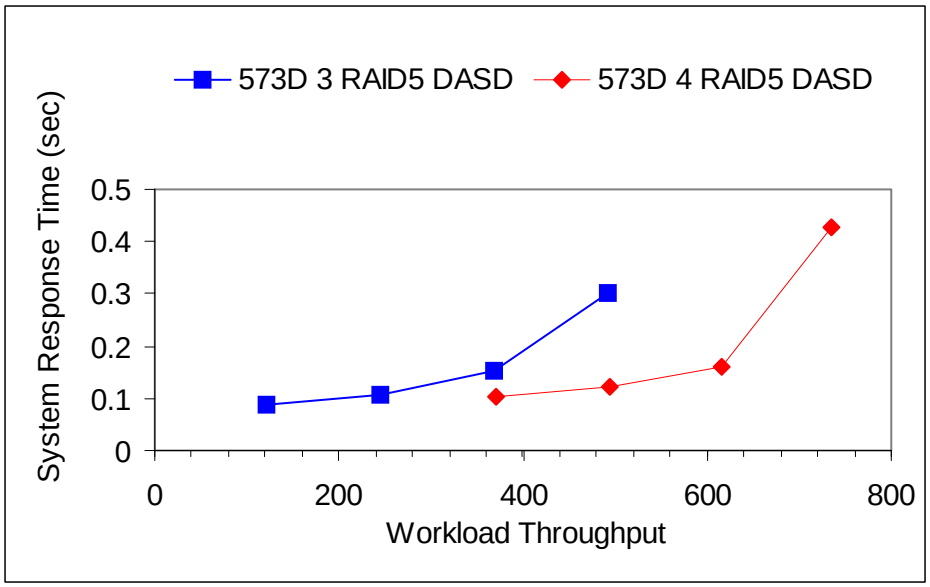
The system CPU % used with and without and IOP was basically the same for the 571B with our workload tests.



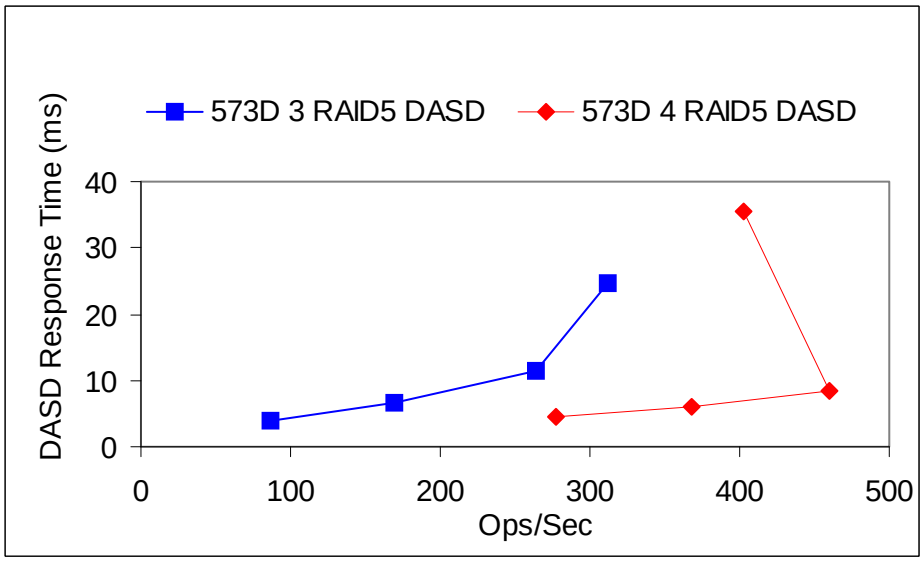
### 14.1.4 571B, 5709, 573D, 5703, 2780 IOA Comparison Chart

In the following two charts we are modeling a System i 520 with a 573D IOA using RAID5, comparing 3 70GB 15K RPM DASD to 4 70GB 15K RPM DASD. The 520 is capable of holding up to 8 DASD but many of our smaller customers do not need the storage. The charts try to point out that there may be performance considerations even when the space isn't needed.

14.1.4.1




14.1.4.2

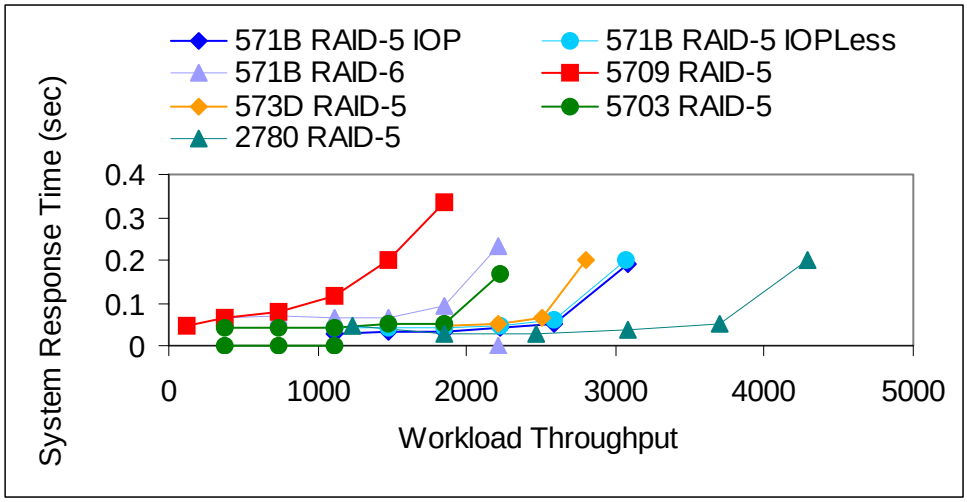


The charts below are an attempt to allow the different IOAs available to be compared on a single chart. An I/O Intensive Workload was used for our throughput measurements. The system used was a 520 model with a single 5094 attached which contained the IOAs for the measurements.

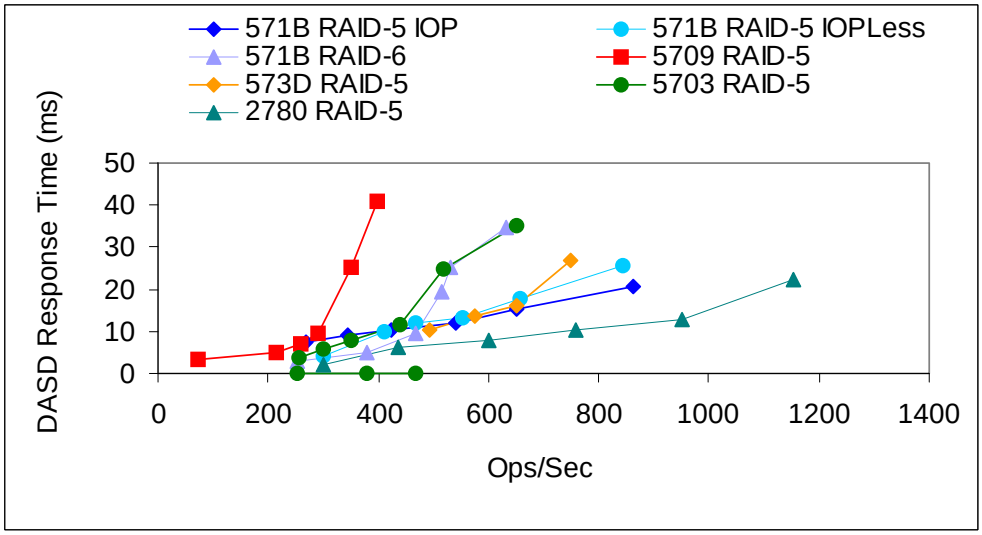
Note: the 5709 and 573D are cache cards for the built in IOA in the 520/550/570 CECs, even though I show them in the following chart like they are the IOA. The 5709 had 8 - 15K 35GB DASD units and the 573D had 8 - 15K 70 GB DASD, the maximum DASD allowed on the built in IOAs.

The other IOAs used 10 - 15K 35GB DASD units. Again this is all for relative comparison purposes as the 571B only supports 10 DASD units in a 5094 enclosure and a maximum of 12 DASD units in a 5095 enclosure. The 2757 and 2780 can support up to 18 DASD units with the same performance characteristics as they display with the 10 DASD units, so when you are considering the right IOA for your environment remember to take into account your capacity needs along with this performance information.

14.1.4.3




14.1.4.4

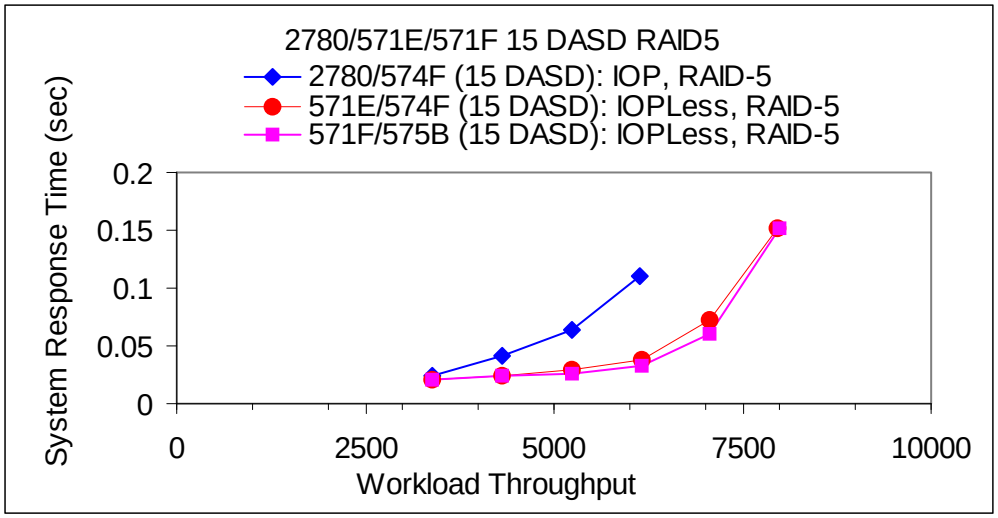


### 14.1.5 Comparing Current 2780/574F with the new 571E/574F and 571F/575B NOTE: iV5R3 has support for the features in this section but all of our performance measurements were done on iV5R4 systems. For information on the supported features see the IBM Product Announcement Letters.

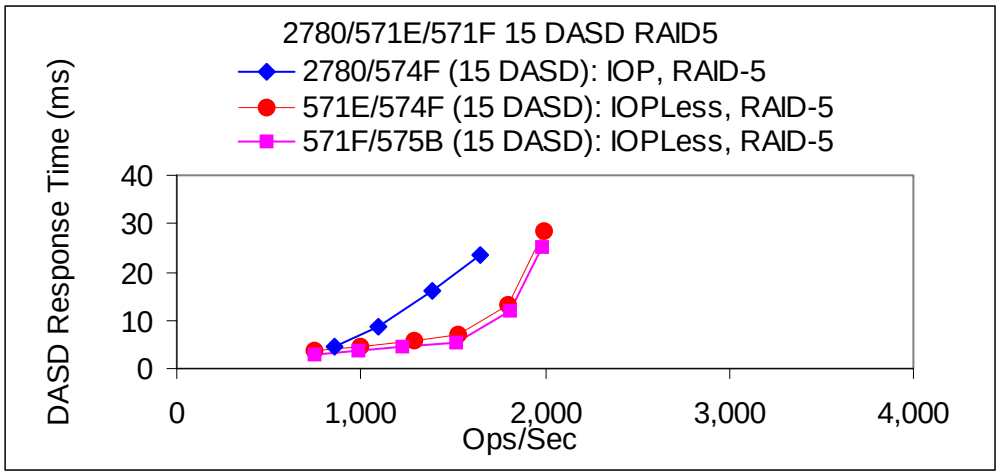
A model 570 4 way system with 48 GB of mainstore memory was used for the following. In comparing the new 571E/574F and 571F/575B with the current 2780/574F IOA, the larger read and write cache available on the new IOA's can have a very positive effect on workloads. Remember this workload is used to get a general comparison between new and current hardware and cannot predict what will happen with all workloads.

Also note the 571E/574F requires the auxiliary cache card to turn on RAID and the 571F/575B has the function included in its double-wide card packaging for better system protection. Understanding of the general results are intended to help customers gauge what might happen in their environments.

14.1.5.1



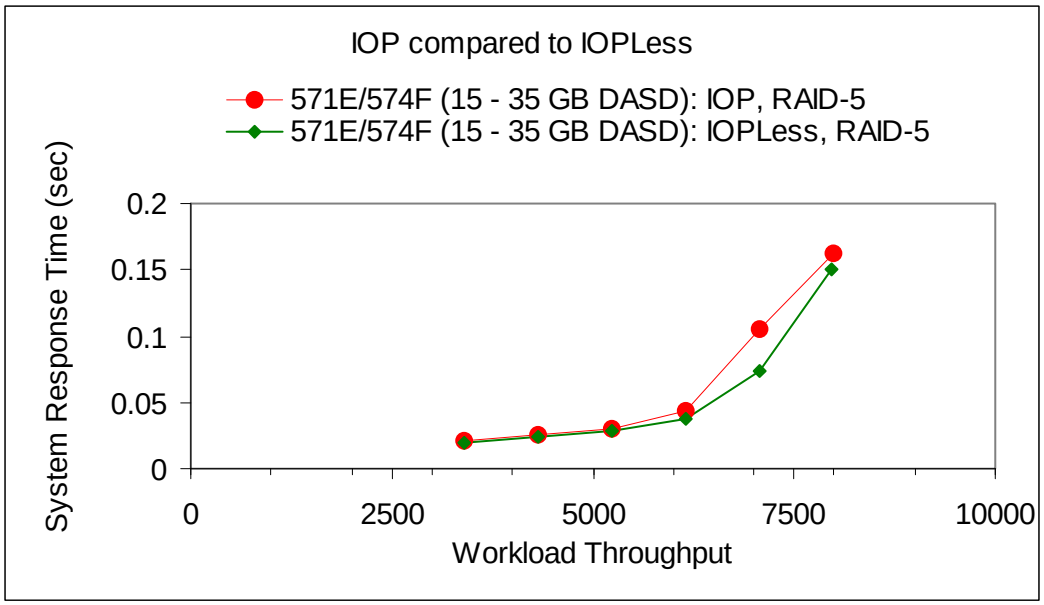

14.1.5.2



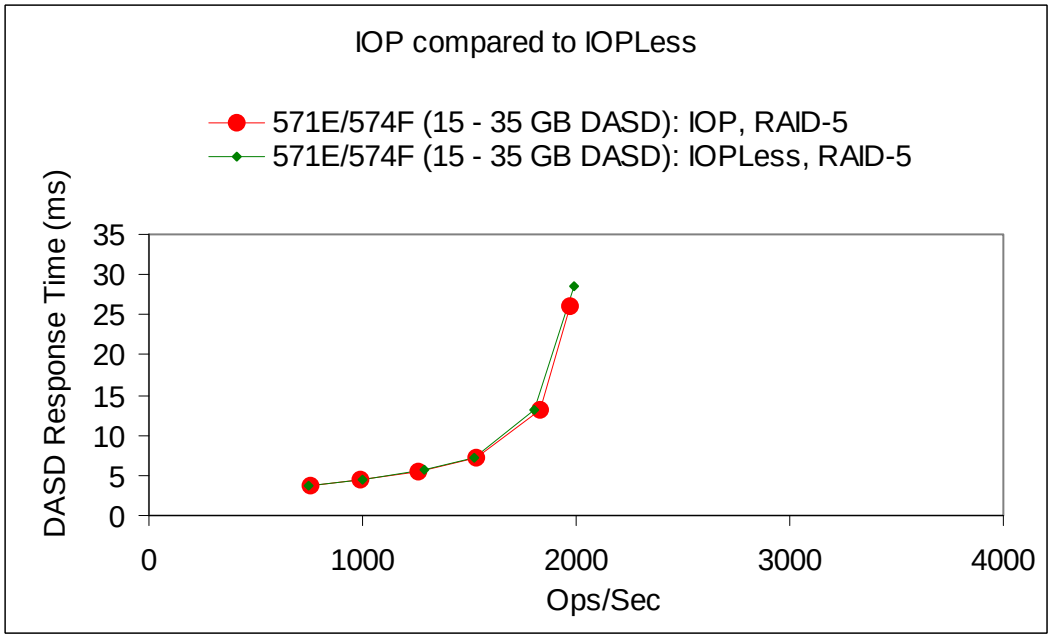
### 14.1.6 Comparing 571E/574F and 571F/575B IOP and IOPLess

In comparing IOP and IOPLess runs we did not see any significant differences, including the system CPU used. The system we used was a model 570 4 way, on the IOP run the system CPU was 11.6% and on the IOPLess run the system CPU was 11.5%. The 571E/574F and 571F/575B display similar characteristics when comparing IOP and IOPLess environments, so we have chosen to display results from only the 571E/574F.

14.1.6.1




14.1.6.2



### 14.1.7 Comparing 571E/574F and 571F/575B RAID5 and RAID6 and Mirroring

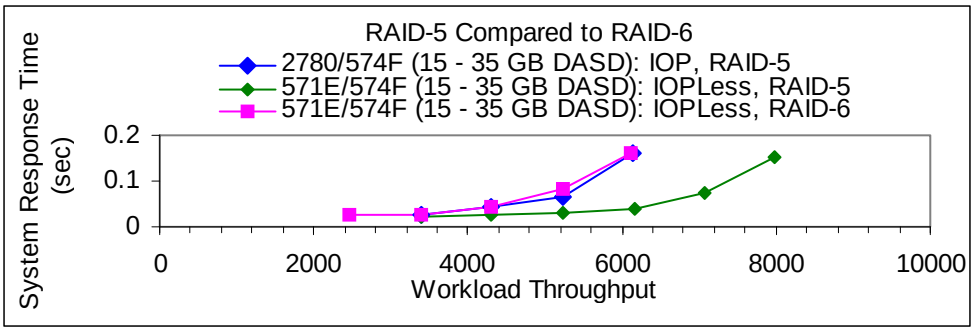
System i protection information can be found at http://www.redbooks.ibm.com/ in the current System i Handbook or the Info Center http://publib.boulder.ibm.com/iseries/ . When comparing RAID5, RAID6 and Mirroring we are interested in looking at the strength of failure protection vs storage capacity vs the performance impacts to the system workloads.

A model 570 4 way system with 48 GB of mainstore memory was used for the following. First comparing characteristics of RAID5 and RAID6; a customer can use Operations Navigator to better control the number of DASD in a RAID set but for this testing we signed on at DST and used default available to turn on our protection schemes. When turning on RAID5 the system configured two RAID sets under our IOA, one with 9 DASD and one with 6 DASD with a total disk capacity of 457 GB. For RAID6 the system created one RAID set with 15 DASD and a capacity of 456 GB. This would generally be true for most customer configurations.

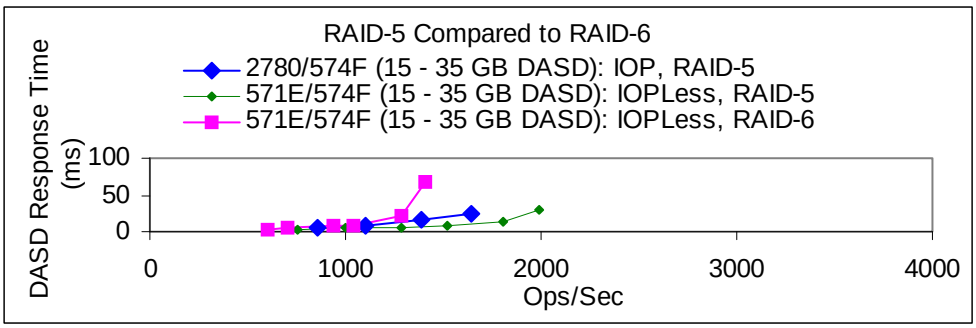
As you look at our run information you will notice that the performance boundaries of RAID6 on the 571E/574F is about the same as the performance boundaries of our 2780/574F configured using RAID5, so better protection could be achieved at current performance levels.

Another point of interest is that as long as a system is not pushing the boundaries, performance is similar in both the RAID5 and RAID6 environments. RAID6 is overwhelmed quicker than RAID5, so if RAID6 is desired for protection and the system workloads are approaching the boundaries, DASD and IOAs may need to be added to the system to achieve the desired performance levels. NOTE: If customers need better protection greater than RAID5 it might be worth considering the IOA level mirroring information on the following page.

14.1.7.1

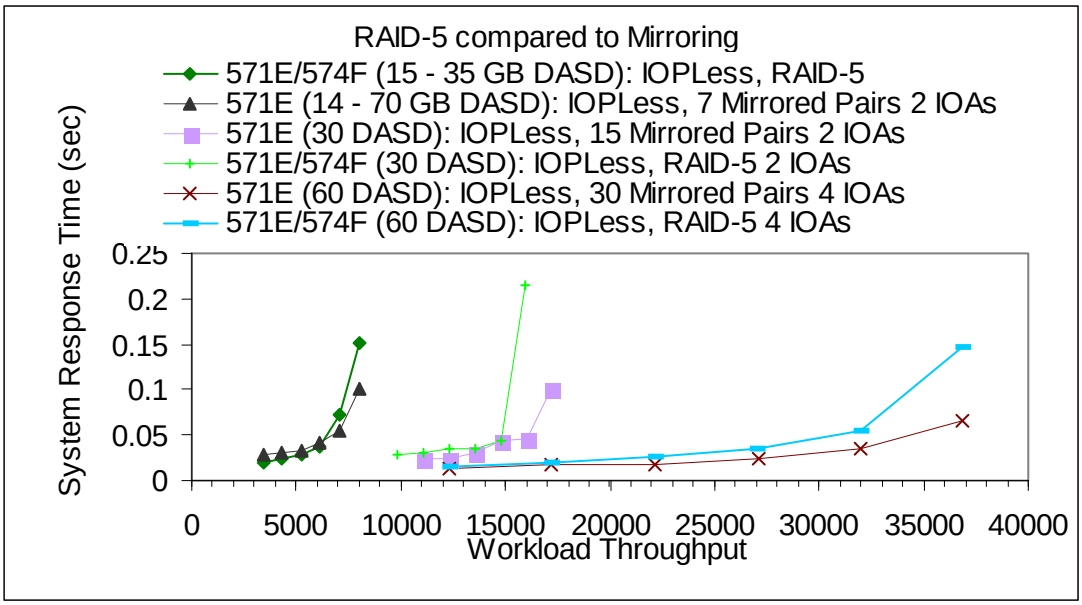



14.1.7.2

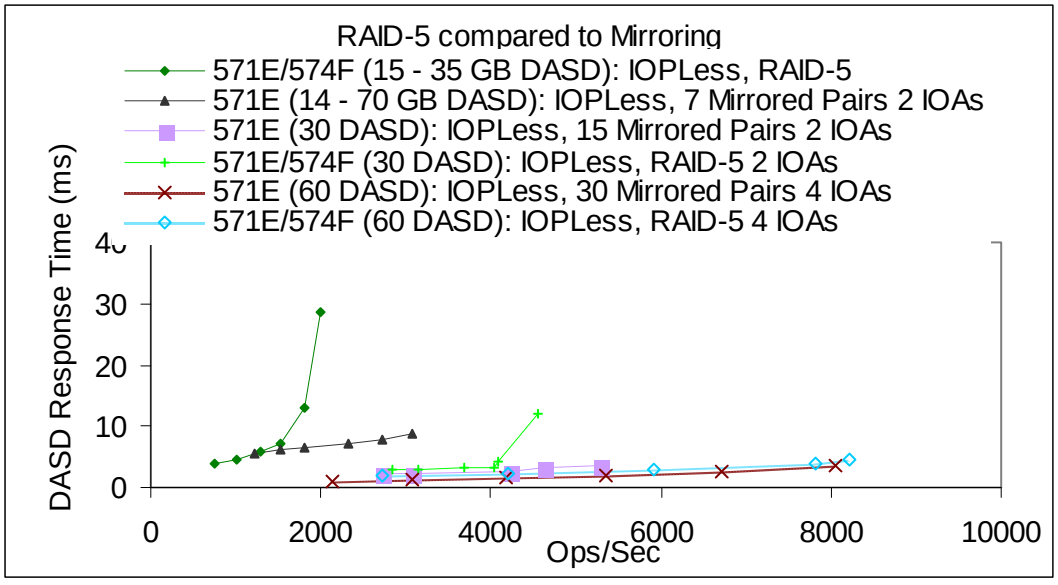


In comparing Mirroring and RAID one of the concerns is capacity differences and the hardware needed. We tried to create an environment where the capacity was the same in both environments. To do this we built the same size database on "15 35GB DASD using RAID5" and "14 70GB DASD using Mirroring spread across 2 IOAs". The protection in the Mirrored environment is better but it also has the cost of an extra IOA in this low number DASD environment. For the 30 DASD and 60 DASD environments the number of IOAs needed is equal.

14.1.7.3




14.1.7.4

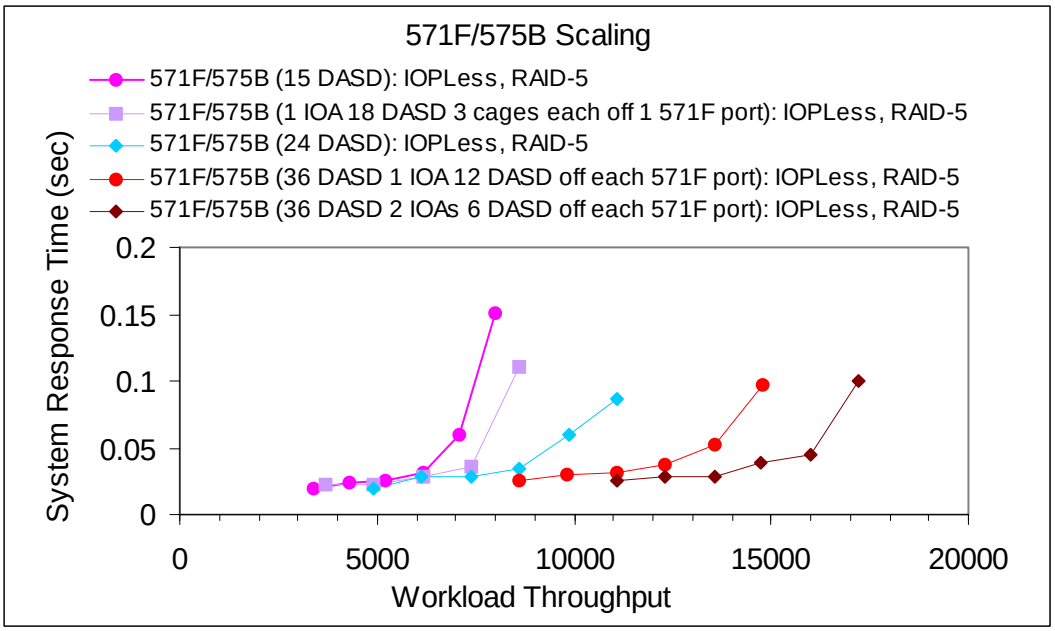


### 14.1.8 Performance Limits on the 571F/575B

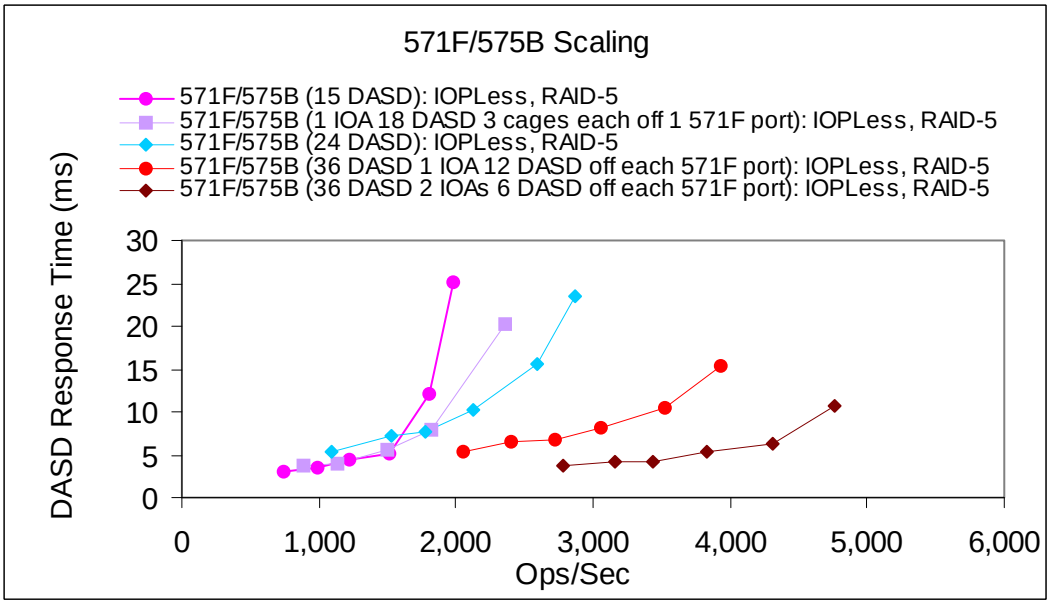
In the following charts we try to characterize the 571F/575B in different DASD configuration. The 15 DASD experiment is used to give a comparison point with DASD experiments from chart 14.1.5.1 and 14.1.5.2. The 18, 24 and 36 DASD configurations are used to help in the discussion of performance vs capacity.

Our DASD IO workload scaled well from 15 DASD to 36 DASD on a single 571F/575B

14.1.8.1




14.1.8.2



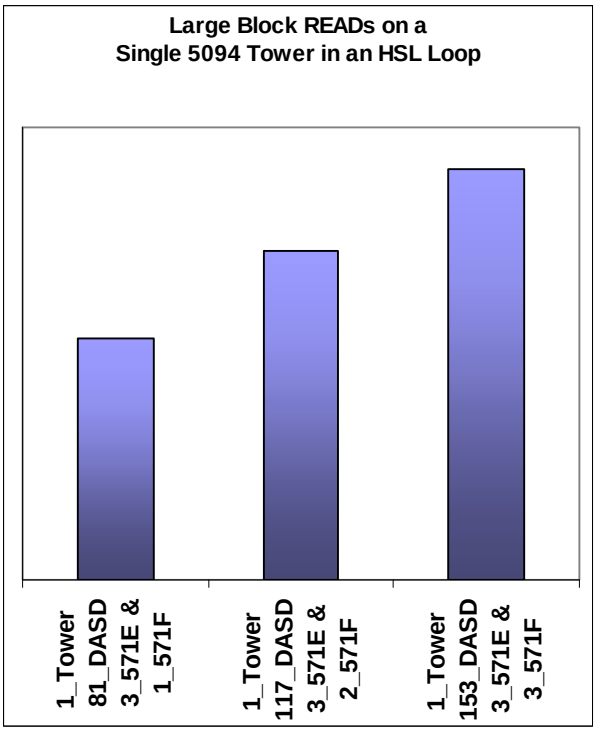
### 14.1.9 Investigating 571E/574F and 571F/575B IOA, Bus and HSL limitations.

With the new DASD controllers and IOPLess capabilities, IBM has created many new options for our customers. Customers who needed more storage in their smaller configurations can now grow. With the ability to add more storage into an HSL loop the capacity and performance have the potential to grow. In the past a single HSL loop only allowed 6 5094 towers with 45 DASD per tower, giving a loop a capacity of 270 DASD, with the new DASD controllers that capacity has grown to 918 DASD. With the new configurations, you can see that 500 and even 600 DASD could make better use of the HSL loop's potential as opposed to the current limit of 270 DASD. Customer environments are unique and these options will allow our customers to look at their space, performance, and capacity needs in new ways.

With the ability to attach so much more DASD to existing towers we want to try to characterize where possible bottlenecks might exist. The first limits are the IOAs and we have attempted to characterize the 571E/574F and 571F/575B in RAID and Mirroring environments. The next limit will be the buses in a single tower. We are using a large file concurrent RSTLIB operations from multiple virtual tape drives located on the DASD in the target HSL loop, to try to help characterize the Bus and HSL limits. The tower is by itself in a single HSL loop, with all the DASD configured into a single user ASP, and RAID5 activated on the IOAs.

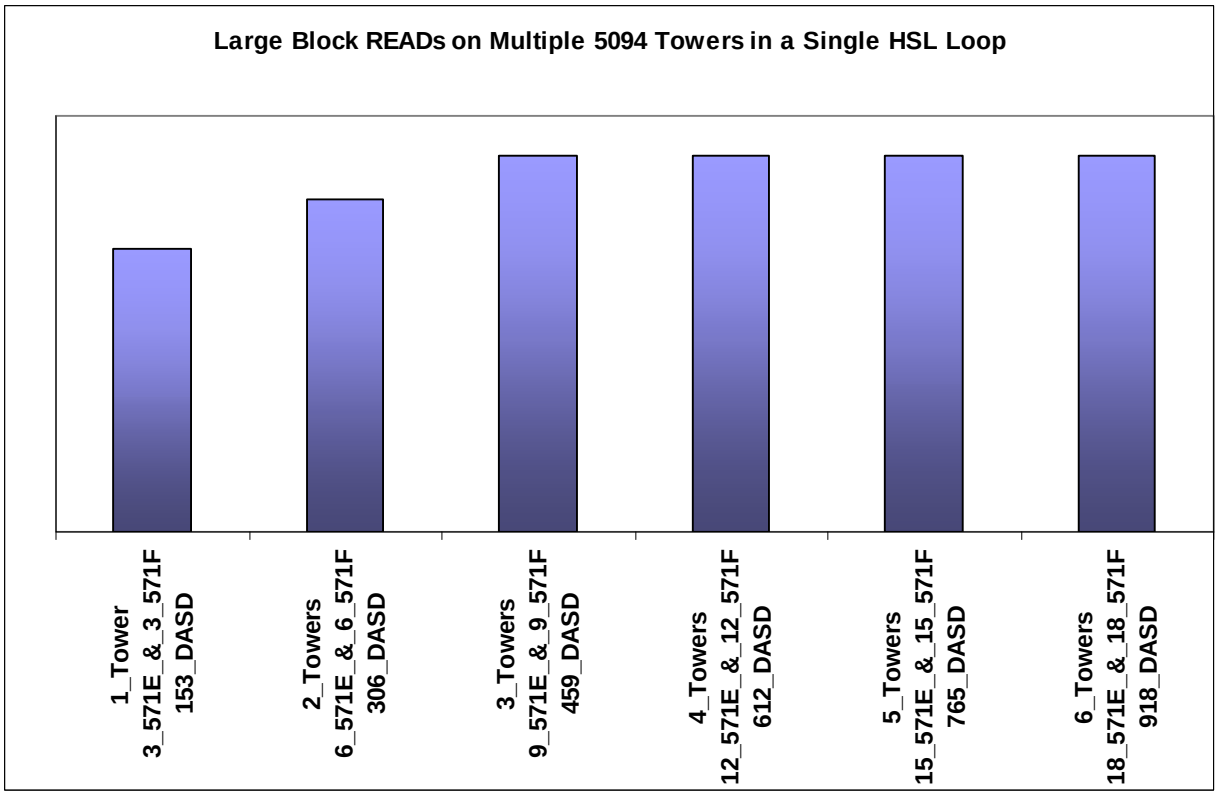
As the scenarios progress 2 then 3 towers are added up to 6 in the HSL loop. All 6 have 3 571E/574F's controlling the 45 DASD in the 5094 towers and 3 571F/575B IOAs controlling 108 DASD in #5786 EXP24 Disk Drawer. Multiple Virtual tape drives were created in the user ASP. The 3 other HSL loops contained the system ASP where the data is written to. We used three HSL loops to prevent the destination ASP from being the bottleneck. The system was a 570 ML16 way with 256 GB of memory and originating ASP contained 916 DASD units on 571E/574F and 571F/575B IOAs. Restoring from the virtual tape would create runs of 100% reads from the ASP on the single loop. The charts show the maximum throughput we were able to achieve from this workload.

NOTE: This is a DASD only workload. No other IOAs such as communication IOAs were present.




14.1.9.1

14.1.9.2



### 14.1.10 Direct Attach 571E/574F and 571F/575B Observations

We did some simple comparison measurements to provide graphical examples for customers to observe characteristics of new hardware. We collected performance data using Collection Services and Performance Explorer to create our graphs after running our DASD IO workload (small block reads and writes).

IOP vs IOPLess: no measurable difference in CPU or throughput.

Newer models of DASD are U320 capable and with the new IOAs can improve workload throughput with the same number of DASD or even less in some workload situations.

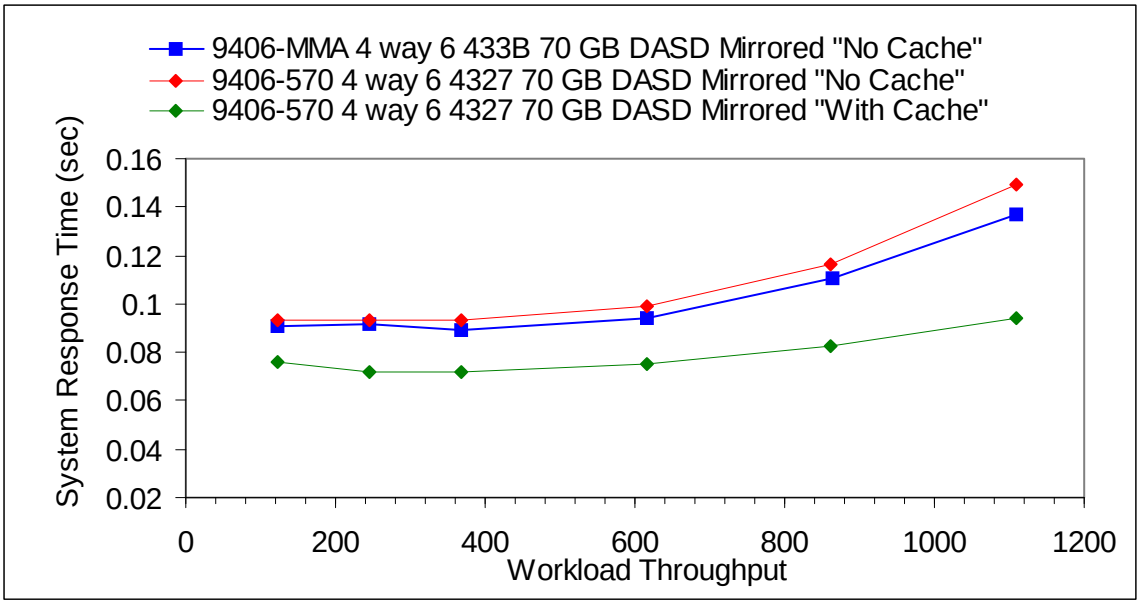
IOA's 571E/574F and the 571F/575B achieved up to 25% better throughput at the 40% DASD Subsystem Utilization point than the 2780/574F IOA. The 571E/574F and 2780/574F were measured with 15 DASD units in a 5094 enclosure. The 571F/575B IOAs attached to #5786 EXP24 Disk Drawers.

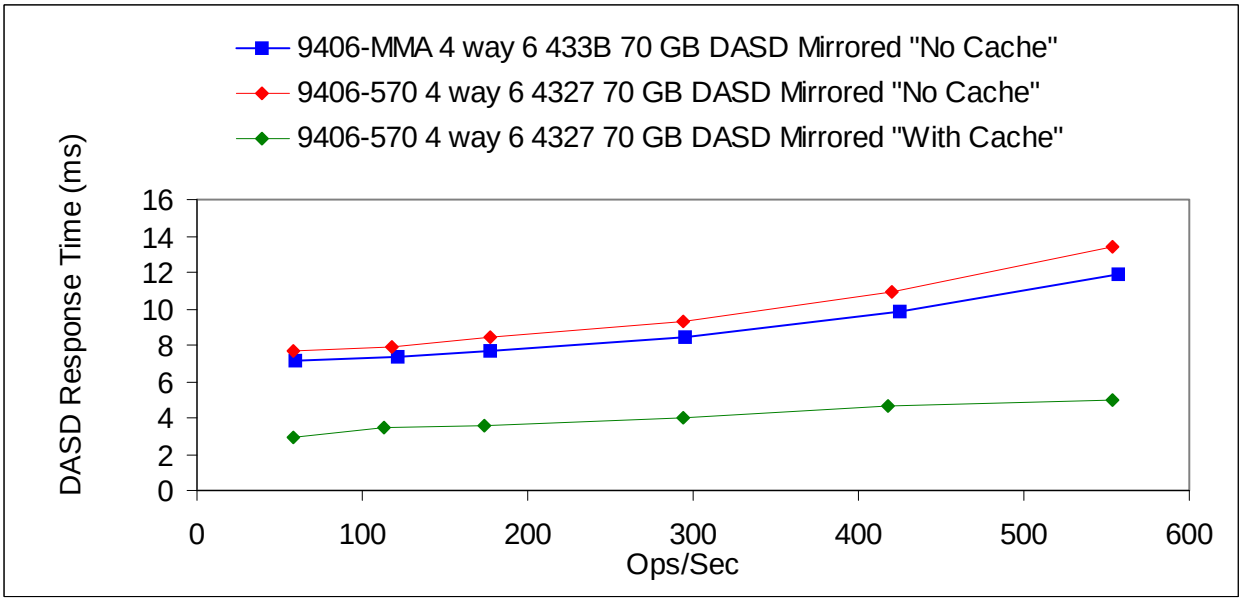
System Models and Enclosures: Although an enclosure supports the new DASD or new IOA, you must ensure the system is configured optimally to achieve the increased performance documented above. This is because some card slots or backplanes may only support the PCI protocol versus the PCI-X protocol. Your performance can vary significantly from our documentation depending upon device placement. For more information on card placement rules see the following link:
IBM i operating system iV5R2: http://www.redbooks.ibm.com/redpapers/pdfs/redp3638.pdf
IBM i operating system iV5R3, iV5R4:
http://www.redbooks.ibm.com/redpapers/pdfs/redp4011.pdf

Conclusions: There can be great benefits in updating to new hardware depending upon the system workload. Most DASD intense workloads should benefit from the new IOAs available. Large block operations will greatly benefit from the 5094/5294 feature code #6417/9517 enclosures in combination with the new IOA's and DASD units.
Note: The #6417/9517 provides a faster HSL-2 interface compared to the #2887/9877 and is available for I/O attached to POWER-5 based systems

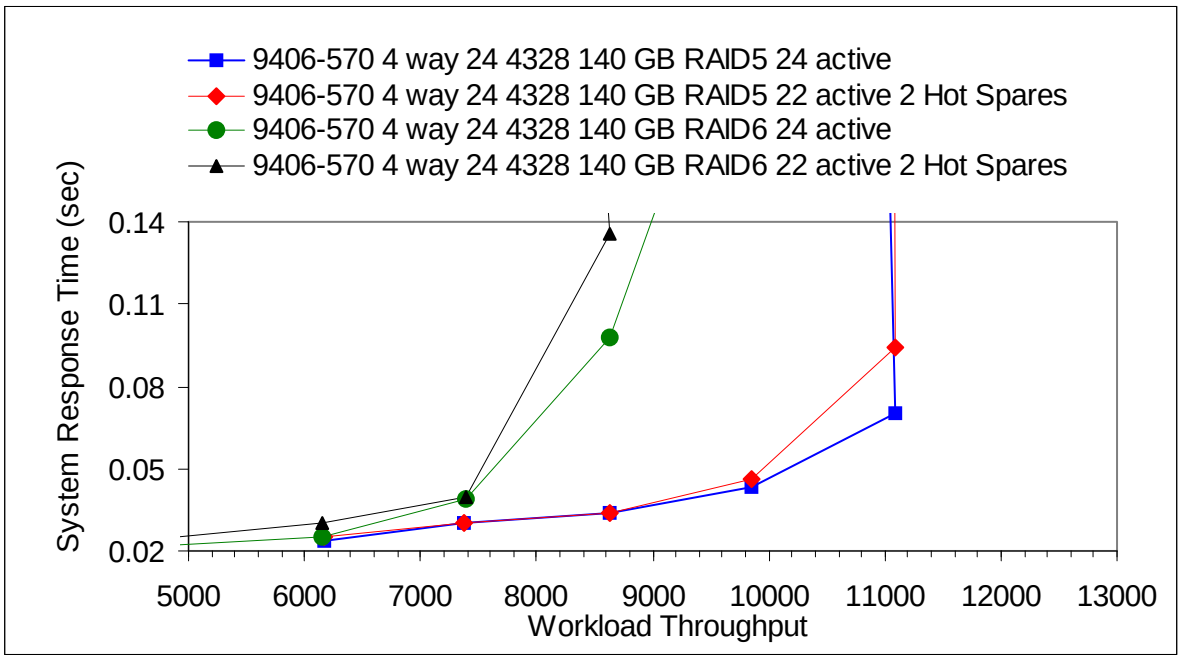
## 14.2 New in iV5R4M5

### 14.2.1 9406-MMA CEC vs 9406-570 CEC DASD

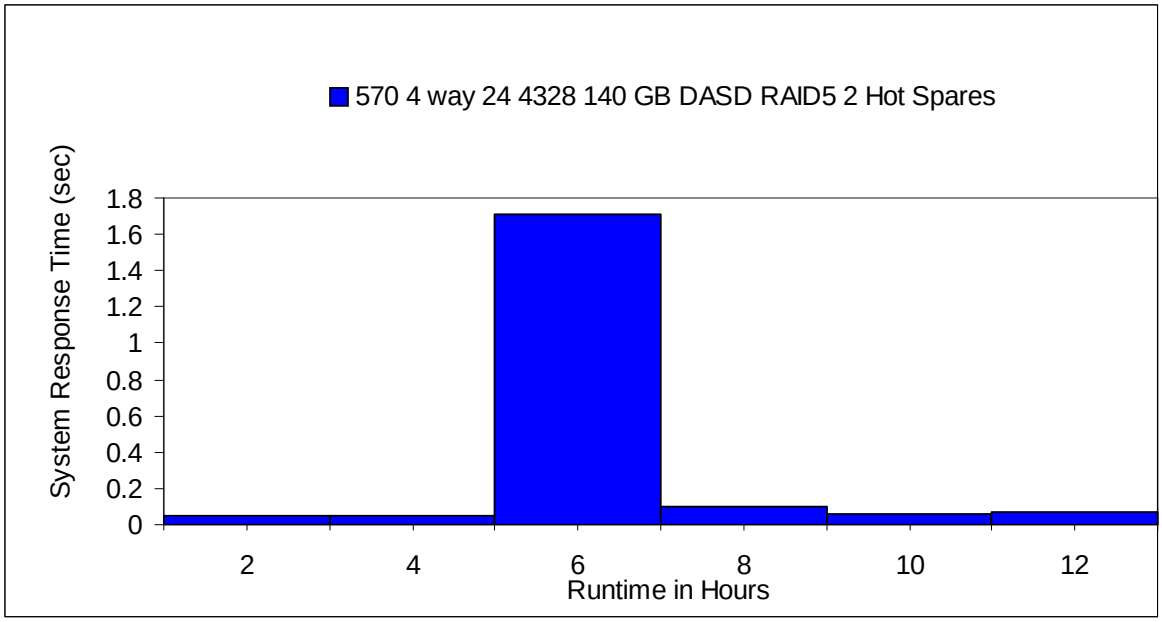


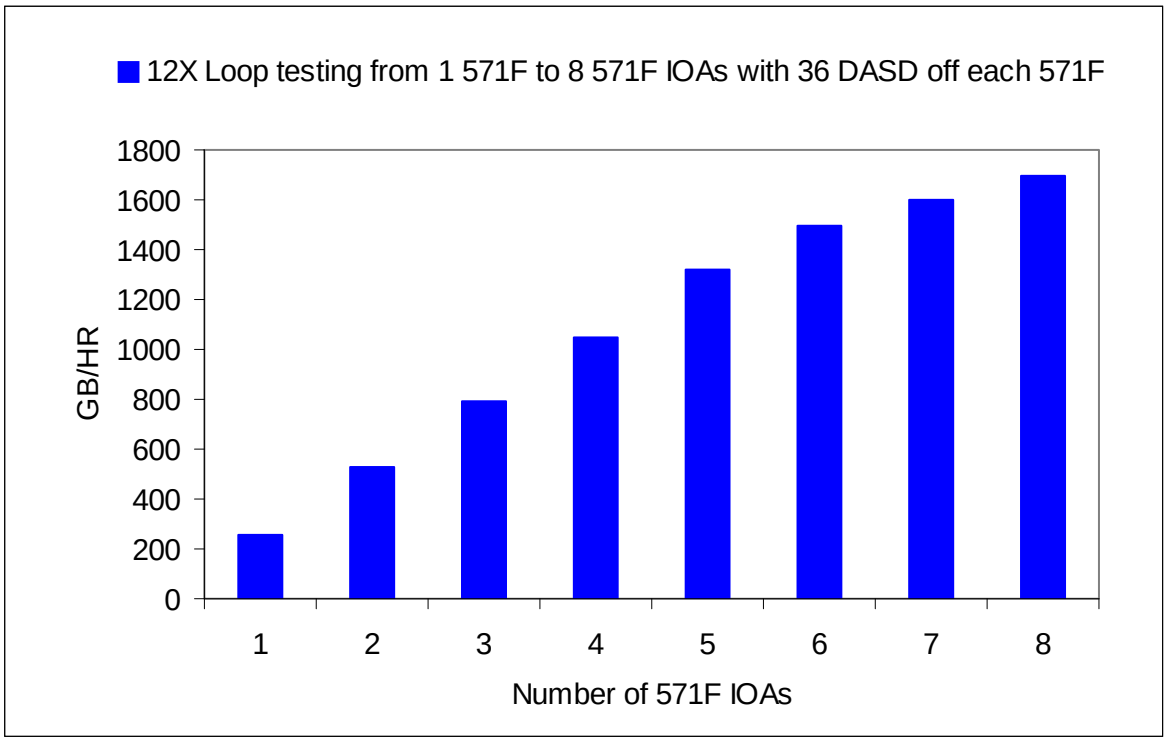
### 14.2.2 RAID Hot Spare




For the following test, the IO workload was setup to run for 14 hours. About 5 hours after starting A DASD was pulled from the configurations. This forced a RAID set rebuild.



### 14.2.3 12X Loop Testing



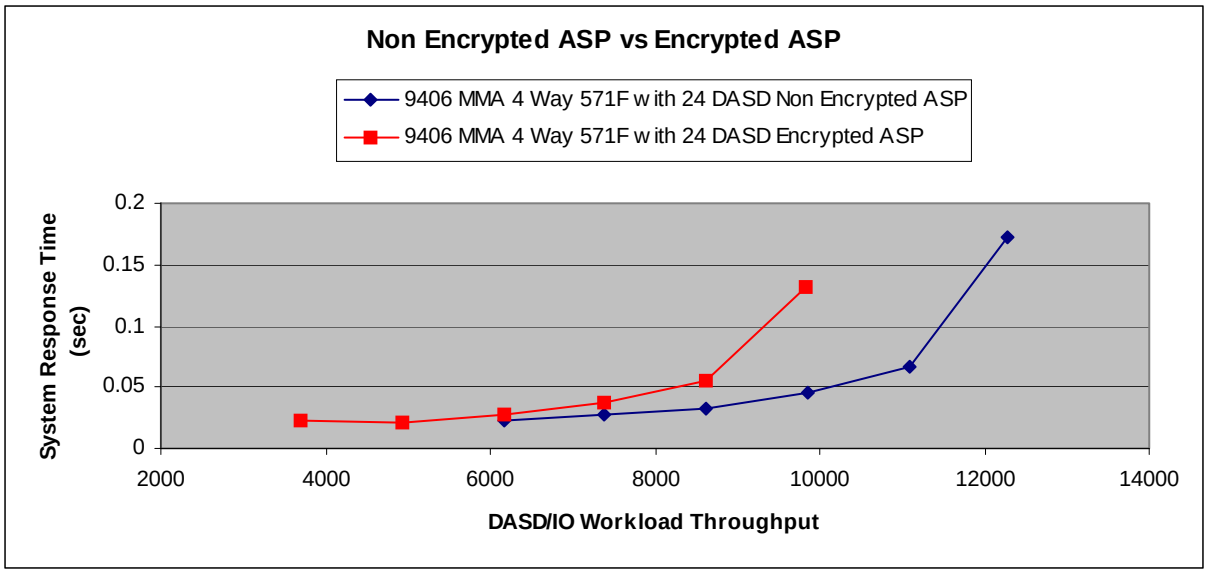

A 9406-MMA 8 Way system with 96 GB of mainstore and 396 DASD in #5786 EXP24 Disk Drawer on 3 12X loops for the system ASP were used, ASP 2 was created on a 4th 12X loop by adding 5796 system expansion units with 571F IOAs attaching 36 4327 70 GB DASD in #5786 EXP24 Disk Drawer with RAID5 turned on. I created a virtual tape drive in ASP2 and I used a 320GB file to save to the tape drive for this test.

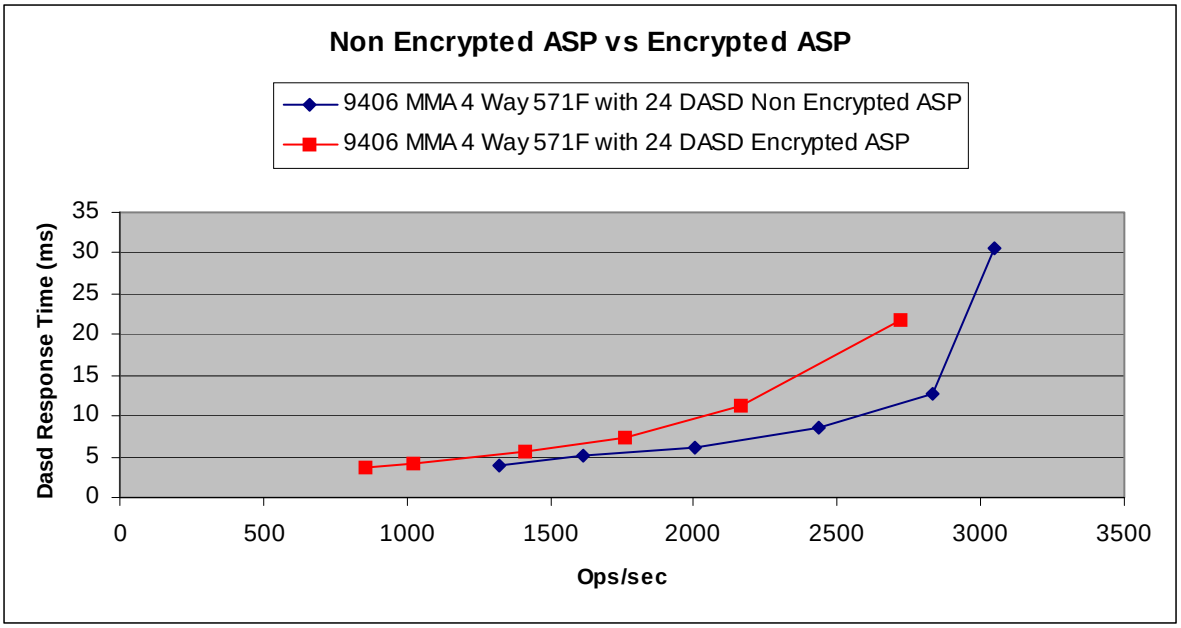
When I completed the testing up to 288 DASD on 8 IOAs, I moved the 12X loop to the other 12X GX adapter in the CEC and ran the test again and saw no difference in the testing between the two loops. The 12X loop is rated for more throughput than the DASD configuration would allow for. So the test isn't a tell all about the 12X loops capabilities only a statement of support to the maximum number of 571F IOAs allowed in the loop.

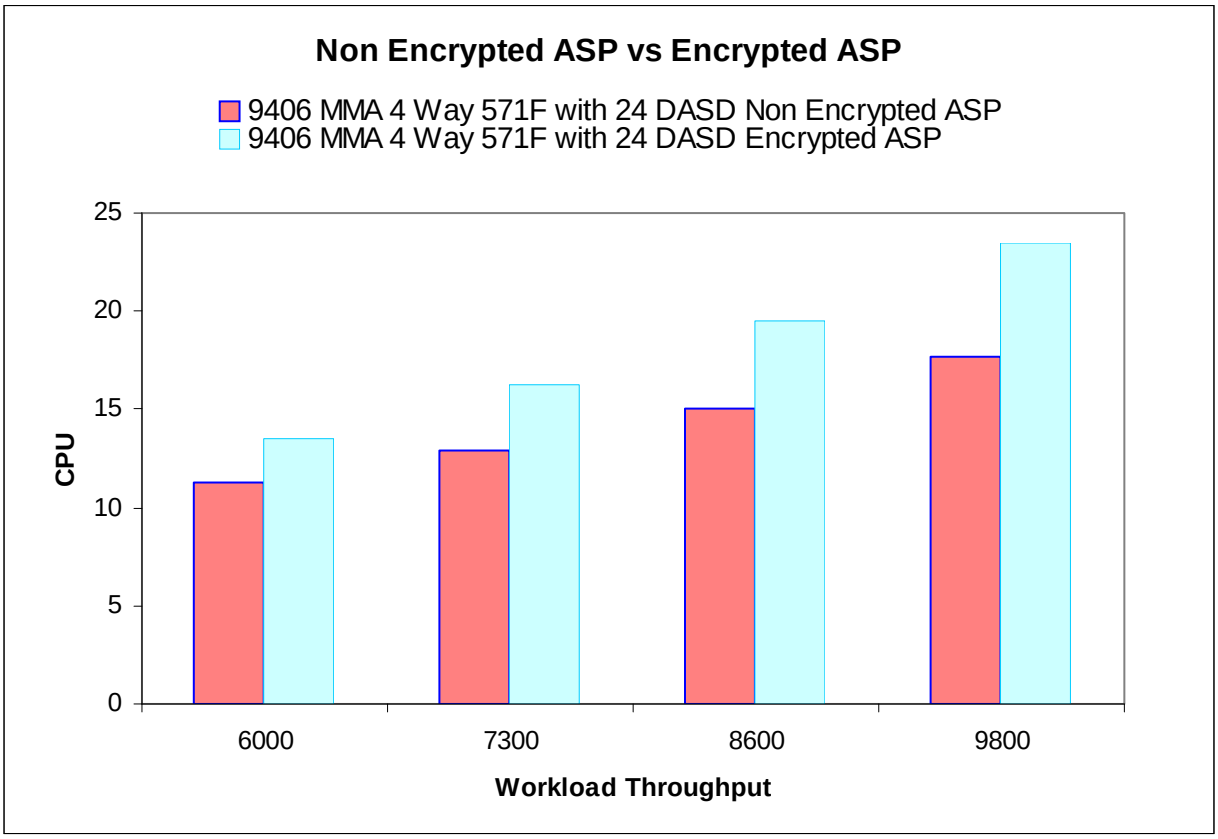
## 14.3 New in iV6R1M0

### 14.3.1 Encrypted ASP

More CPU and memory may be needed to achieve the same performance once encryption is enabled.

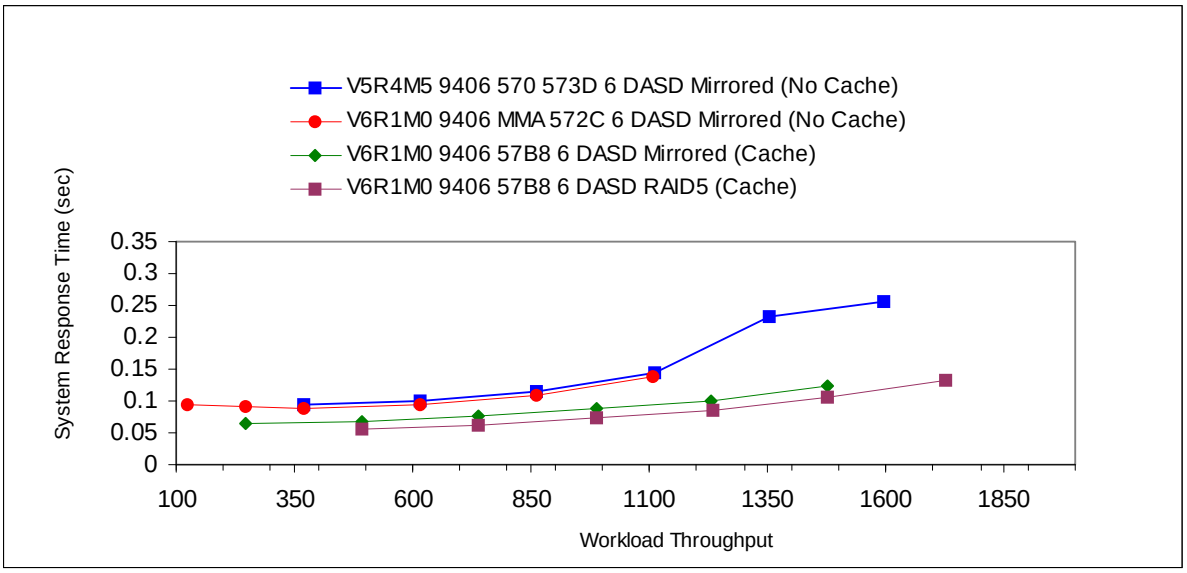


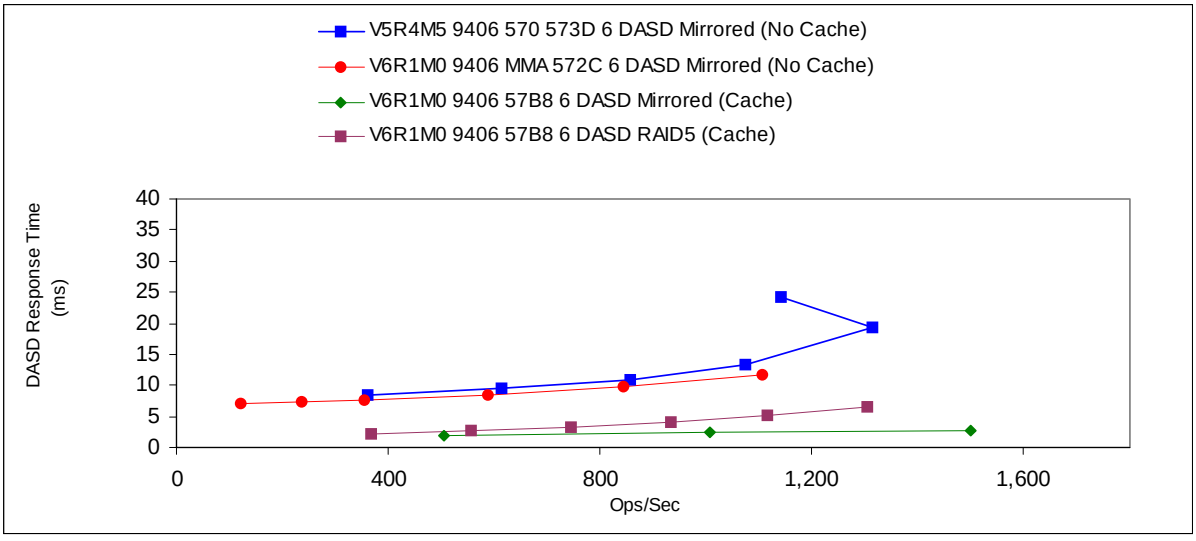





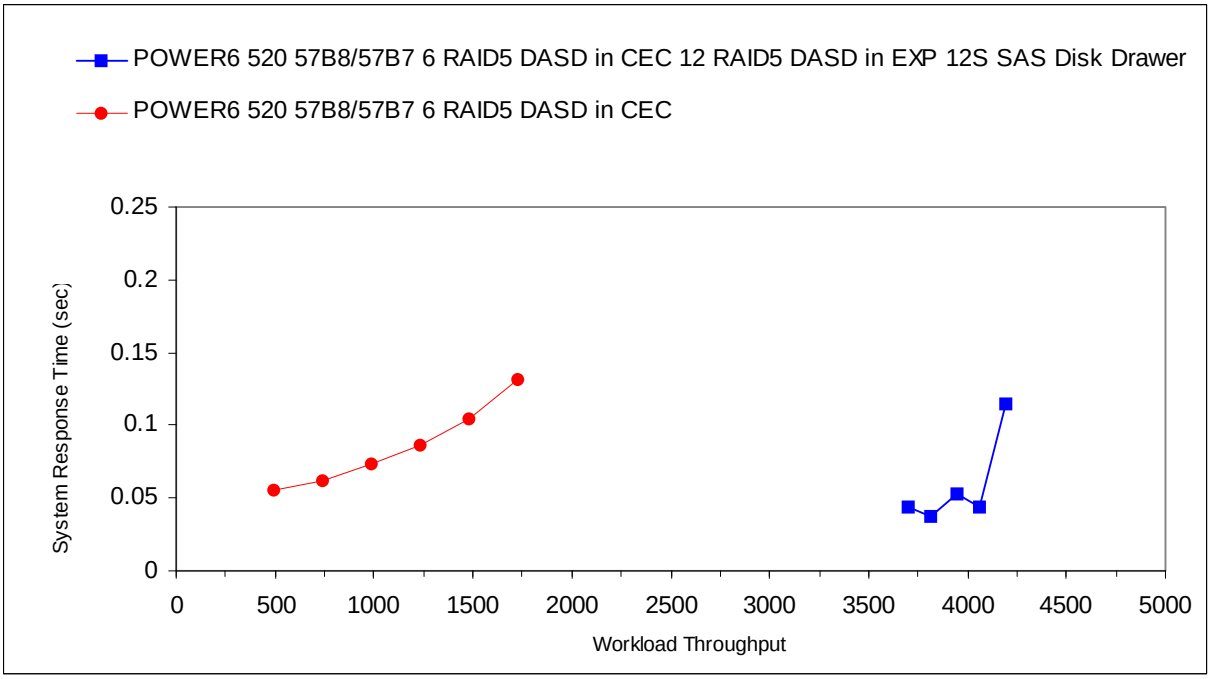
### 14.3.2 57B8/57B7 IOA

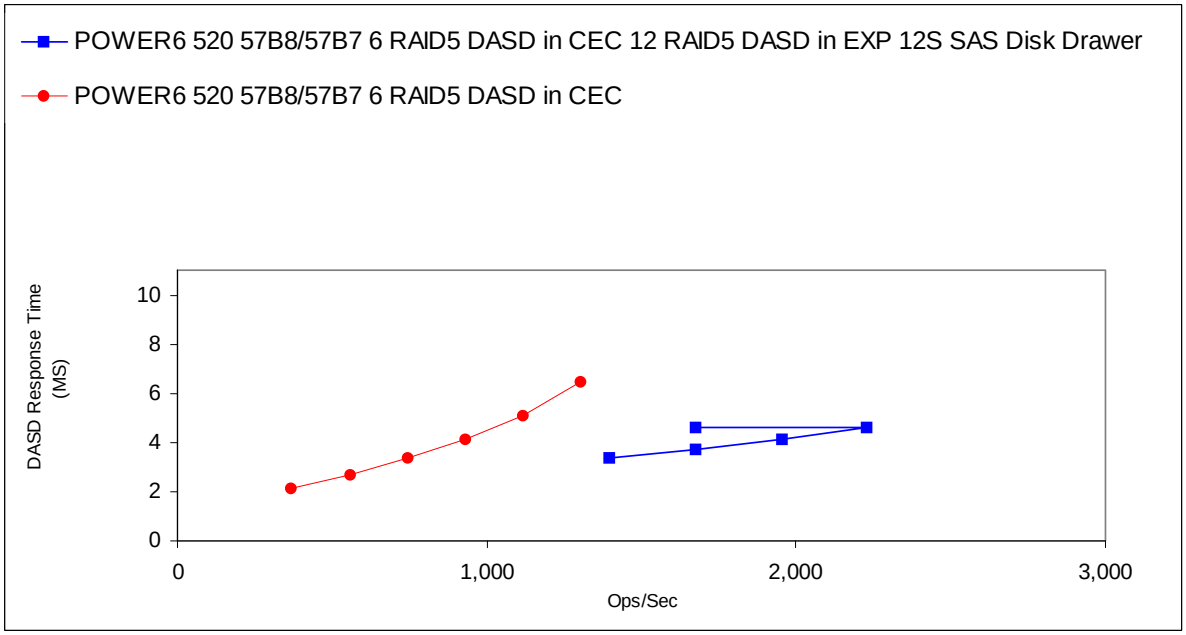
With the addition of the POWER6 520 and 550 systems comes the new 57B8/57B7 SAS Raid Ennoblement Controller with Auxiliary Write Cache. This controller is only available in the POWER6 520 and 550 systems and provides RAID5/6 capabilities, with 175MB redundant write cache. Below are some charts comparing the Storage Controllers for the POWER5 570 (573D), which can be either mirrored or RAID5 protected. The POWER6 570 (572C) which can only be mirrored, and the POWER6 520/550 (57B8/57B7) which can be RAID5/6 or protected with mirroring.



The POWER6 520 and 550 also have an external SAS port, that is controlled by the 57B8/57B7, used to connect a single #5886 - EXP 12S SAS Disk Drawer which can contain up to 12 SAS DASD. Below is a chart showing the addition of the #5886 - EXP 12S SAS Disk Drawer.



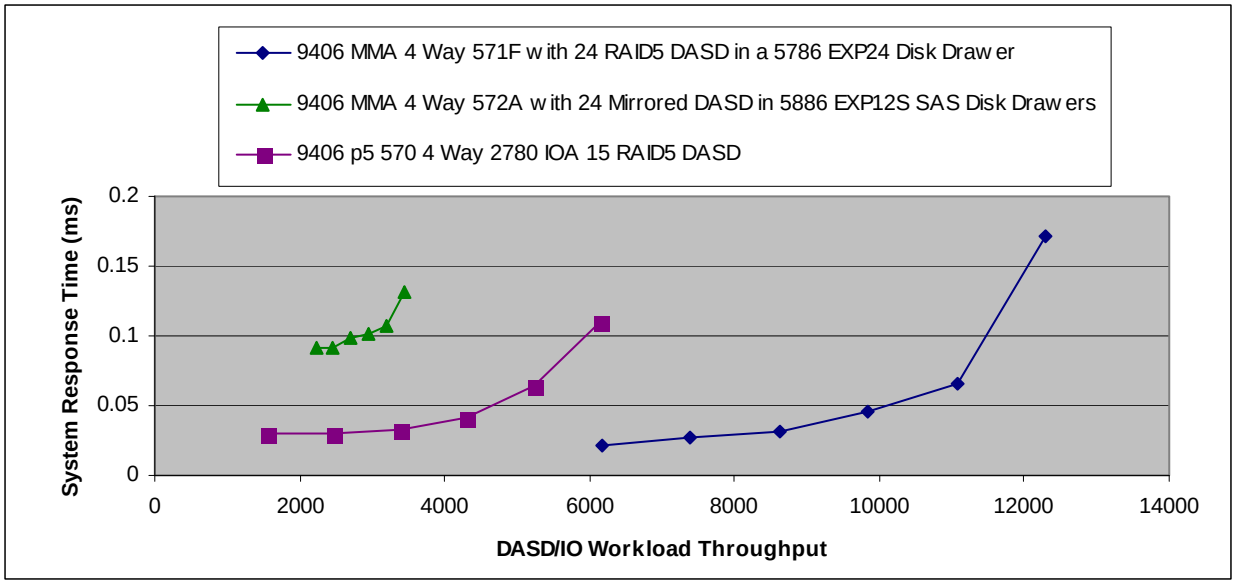
### 14.3.3 572A IOA

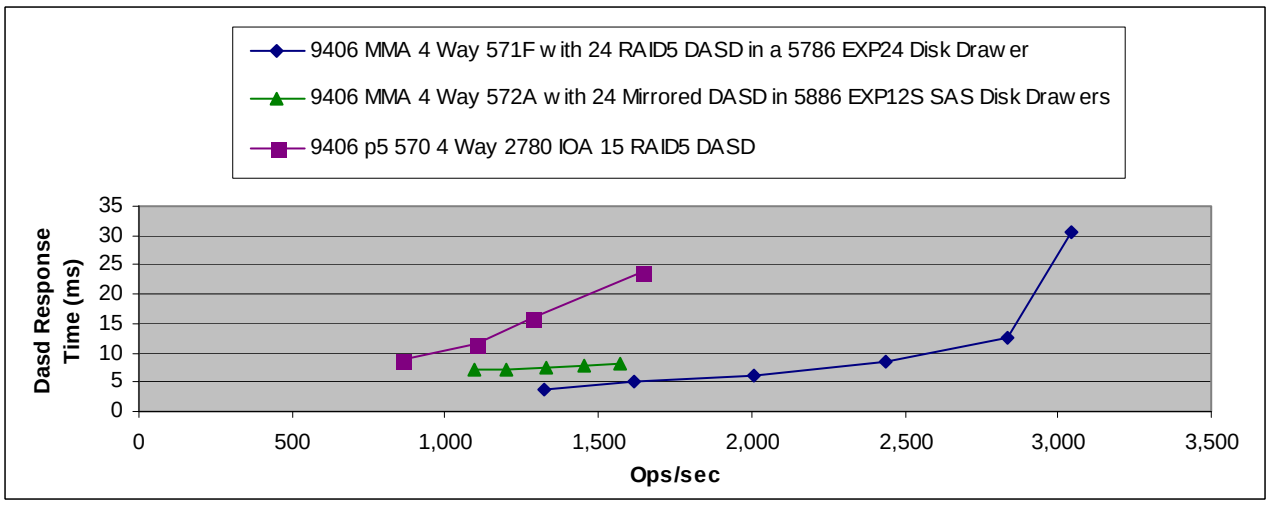
The 572A IOA is a SAS IOA that is mainly used for SAS tape attachment but the 5886 EXP 12S SAS Disk Drawer can also be attached.

Performance will be poor as the IOA does not have any cache.

The following charts help to show the performance characteristics that resulted during experiments in the Rochester lab.

If storage space is all that is needed then the 5886 EXP 12S SAS Disk Drawer could be an option



## 14.4 SAN - Storage Area Network (External)

There are many factors to consider when looking at external storage options, you can get more information through your IBM representative and the white papers that are available at the following location.

https://www-304.ibm.com/systems/support/

## 14.5 iV6R1M0 -- VIOS and IVM Considerations

Beginning in iV6R1M0, IBM i operating system will participate in a new virtualization strategy by becoming a client of the VIOS product. Customers will view the VIOS product two different ways:

- On blade products, through the regular configuration tool IVM (which includes an easy to use interface to VIOS).
- On traditional (non-blade) products, through a combination of HMC and the VIOS command line.

The blade products have a simpler interface which, on our testing, appear to be sufficient for the environments involved. On blades, customers are restricted to a single set of 16 logical units (which IBM i operating system perceives as if they were physical drives). This substantially reduces the number and value of tuning options. It is possible for blade-based customers to use the VIOS command line. However, we did not discover the need to do so and do not think most customers will need to either. The tuning available from IVM proved sufficient and should be preferred for its ease of use when it is workable.

Customers should strongly consider their disk requirements here and consult with their support teams before ordering. Customers with more sophisticated disk-based requirements (or, simply, larger numbers of disks) should choose systems that allow a greater number of LUNs and thereby enable more substantial tuning options provided from the VIOS command line. No hard and fast rules are possible here and we again emphasize that one consult with their support team on what will work for them. However, as a broad rule of thumb, customers with 200 or more physical drives very likely need something beyond the 16 LUNs provided by the IVM environment. Customers below 100 physical disks can, in most cases, get by with IVM. Customers with 50 or fewer very likely will do just fine with IVM.

## 14.5.1 General VIOS Considerations

### 14.5.1.1 Generic Concepts

***520 versus 512*.** Long time IBM i operating system users know that IBM i operating system disks are traditionally configured with 520 byte sectors. The extra eight bytes beyond the 512 used for data are used for various purposes by Single Level Store.

For a variety of reasons, VIOS will always surface 512 byte sectors to IBM i operating system whatever the actual sector size of the disk may be. This means that 520 byte sectors must be emulated within 512 byte sectors when using disks supported by VIOS. This is done, simply enough, by reading nine 512 byte data sectors for every eight sectors of actual data and placing the Single Level Store information within the extra sector. Since all disk operations are controlled by Single Level store in an IBM i operating system there are no added security implications from this extra sector, provided standard, sensible configuration practices are followed just as they would be for regular 520 byte devices.

However, reading nine sectors when only eight contain data will cost some performance, most of it being the sheer cost of the extra byte transfer of the extra sector. The gains are the standard ones of virtualization -- one might be able to share or re-purpose existing hardware for System i's use in various ways.

Note carefully that some "512" byte sectored devices actually have a range of sizes like 522, 524, and others. Confusingly for us, the industry has gone away from strictly 512 byte sectors for some devices. They, too, have headers that consume extra bytes. However, as noted above, these extra bytes are not available for IBM i operating system and so, for our purposes, they should be considered as if they were 512 byte sectored, because that is what IBM i operating system will see. Some configuration tools, however, will discuss "522 byte" or whatever the actual size of the sectors is in various interfaces (IVM users will not see any of this).

***VIOS will virtualize the devices***. Many configuration options are available for mapping physical devices, as seen by VIOS, to virtual devices that VIOS will export to DST and Single Level Store. Much more of this will be done by the customer than was done with internal disks. Regardless of whether the environment is blades or traditional, it is important to make good choices here. Even though there is much functional freedom, many choices are not optimized for performance or optimized in an IBM i operating system context. Moreover, nearly as a matter of sheer physics, some choices, once made, cannot be much improved without very drastic steps (e.g. dedicating the system, moving masses of data around, etc.). Choosing the right configuration in the first place, in other words, is very important. Most devices, especially SAN devices, will have “Best Practices” manuals that should be consulted.

### 14.5.1.2 Generic Configuration Concepts

There are several important principles to keep track of in terms of getting good performance. Most of the following are issues when the disks are configured. A great many problems can be eliminated (or, created) when the drives are originally configured. The exact nature of some of these difficulties might not be easily predicted. But, much of what follows will simply avoid trouble at no other cost.

1. ***Ensure that RAID protection is performed as close to the physical device as possible .*** This is typically done out at an I/O adapter level or on the external disk array product. This means that either the external disk's configuration tools or (for internal disks assigned to VIOS) VIOS' tools will be used to create RAID configurations (RAID5, RAID10, or RAID1). When this is done, as far as IBM i operating system disk status displays are concerned, the resulting virtual drives appear to be "unprotected." It might be superficially reassuring to have IBM i operating system do the protection (if IBM i operating system even permits it). WRKDSKSTS would then show the protection on that path. DST/SST disk configuration functions would show the protection, too. However, it is better to put up with what appears to IBM i operating system's disk status routines to be unprotected devices (which are, after all, actually protected) than to take on the performance problems of doing this under IBM i operating system. RAID recovery procedures will have to be pursued outside of IBM i operating system in any event, so the protection may as well go where the true physicality is understood (either in VIOS or the external disk array product).
Note also that you also want to configure things so that the outboard devices, rather than VIOS, do the RAID protection whenever possible. This enables I/O to flow directly from the device to IBM i operating system as directed by VIOS.

High Availability scenarios also need to be considered. In some cases, to enable appropriate redundancy, it may be necessary to do the protection a little farther away from the device (e.g. spread over a couple of adapters) so as to enable the proper duplexing for high availability. If this applies to you, consult the documentation. Some external storage devices have extensive duplexing within themselves, for instance, which could allow one to keep the protection close to the device after all.

2. ***Recognize that Internal Disks remain the "gold standard" for performance.*** We have consistently measured external disks as having less performance than 520 byte, internally attached disks. However, the loss of throughput, with proper configuration, is not a major concern. What is harder to control is response time. If you have sensitivity to response time, consider internal disks more strongly.

3. ***Prefer external disks attached directly to IBM i operating system over those attached via VIOS*** This is basically a statement of the Fibre Channel adapter and who owns it. In some cases, it affects which adapter is purchased. If you do not need to share a given external disk's resources with non-IBM i operating system partitions, and the support is available, avoiding VIOS altogether will give better performance. First, the disks will usually have 520 byte support. Second, the IBM i operating system support will know the device it is dealing with. Third, VIOS will typically run as a separate partition. If you run VIOS as your first shared partition, simply turning on shared support costs about five to eight percent overall. The alternative, a dedicated partition for VIOS, would be a nice thing to avoid if possible. If you would not have used shared processor support otherwise, or would have to give VIOS a whole processor or more otherwise, this is a consideration.

4. ***Prefer standard IBM i operating system internal disks to VIOS internal disks***. This describes who should own a given set of internal disks. If there is a choice, giving the available internal disks to IBM i operating system instead of going through VIOS will result in noticeably better performance. VIOS is a better fit for external disk products that do not support the IBM i operating system 520 byte sector. The VIOS case would include internal disks that came originally from pSeries or System p. However, one should investigate those devices also. If those devices support 520 byte sectors (or, alternatively, if it is stated they are supported by IBM i operating system), they should be reconfigured instead as native IBM i operating system internal disks. It should be exceptional to use VIOS for internal disks.

5. ***Prefer RAID 1 or RAID 10 to RAID 5***. We are now beginning to generally recommend RAID 1 ("mirroring") or RAID 10 (a "mirroring" variant) for disks generally in On-line Transaction Processing (OLTP) environments. OLTP environments have long had to deal with configurations based on total arm count, not capacity as such. If that applies to you, you have extra space that is of marginal value. Those in this situation can nowadays use the same number of arms deployed as RAID 1 or RAID 10 to gain increased performance. This is at least as true for external disks as it is for internal disks. Note that in this recommendation, one deploys the same arm count -- just deploys them differently, trading unused space for performance. Also note that if one goes this route, two physical disks per RAID 10 or RAID 1 set is better than a larger number of disks per RAID 1 or RAID 10 set. (See also “Ensure, within reason” below).

6. ***For VIOS, Prefer External Disks (SAN disks) to Internal Disks***. SAN disks will have greater flexibility and better tuning options than internal disks. Accordingly, when there is a choice, VIOS is best used for external disks.

7. ***Separate Journal ASPs from other ASPs.*** Generally, we have long recommended that a given set of data base files (aka SQL tables) keep its set of journal receivers in a separate ASP from the data base ASP or ASPs. With VIOS, we recommend that this continue to the extent feasible. It may be necessary to share things like Fibre Channel links, but it should be possible to have separate physical devices at the very least. To the extent possible, arrange for journal to use its own internal buses also (of whatever sort the device provides).

8. ***Ensure, within reason, a reasonable number of virtual disks are created and made available to IBM i operating system.*** One is tempted to simply lump all the storage one has in a virtual environment into a couple (or even one) large virtual disk. Avoid this if at all possible.

For traditional (non-blade) systems: There is a great deal of variability here, so generalizations are difficult. However, in the end, favor virtual disks that are within a binary order of magnitude or two of the physical disk sizes. Make each them as close to the same size if possible. In any case, strive to have half a dozen or more in an ASP if you can. Years of system tuning (at all levels) tacitly expect a reasonable number of devices, so it makes sense to provide a bunch. You don't need a count larger than the physical device count, however, unless the device count is very small.

For blades-based systems: You only have 16 LUNs available. However, you should use a good fraction of them rather than merely one or two. In our tests, we tended to use twelve to sixteen LUNs. One wishes a sufficient number for IBM i operating system to work with -- one wishes also to segregate physical devices between ASPs to the extent feasible.

9. ***Prefer Symmetrical Configurations***. To the extent possible, we have found that physical symmetry pays off more than we have seen before. Balancing the number of physical disks as much as possible seems to help. Strive to have uniform LUN sizes, uniform number of disks in each RAID set, balance (at least at the static configuration level) between the various internal and external buses, etc. To the extent practical, the user should strive for even numbers of items.

10. ***In general, do not share the same physical disk with multiple partitions.*** Only If you are running some minimal IBM i operating system partition (say, a very small Domino partition or perhaps a middle tier partition that has no local data base), should you consider strategies where IBM i operating system is sharing physical disks with other partitions. For more traditional application sets (whether a traditional system or a blade) you'll have a data base or large enough data contents generally to give each IBM i operating system partition its own physical devices. Once you get to multiple devices, sharing them with other partitions will lead to performance problems as the two partitions fight (in mutual ignorance) for the same arm, which may increase seek time (at least) a little to a lot. Service time could be adversely affected as well.

11. ***To the extent possible, think multiple VIOS partitions for multiple IBM i operating system partitions.*** If the physical disks deserve segmentation, multiple VIOS partitions may also be justified. The main issue is load. If the IBM i operating system partitions are small (under two CPUs), then you're probably better off with a shared VIOS partition hosting a couple of small IBM i operating system partitions. As the IBM i operating system partitions grow, it will be possible to justify dedicated VIOS partitions. Our current measurements suggest one VIOS processor for every three IBM i operating system processors, but this will vary by the application.

### 14.5.1.3 Specific VIOS Configuration Recommendations -- Traditional (non-blade) Machines

1. ***Avoid volume groups if possible***. VIOS "hdisks" must have a volume identifier (PVID). Creating a volume group is an easy way to assign one and some literature will lead you to do it that way. However, the volume group itself adds overhead for no particular value in a typical IBM i operating system context where physical volumes (or, at least, RAID sets) are exported as a whole without any sort of partitioning or sub-setting. Volume groups help multiple clients share the same physical disks. In an IBM i operating system setting, this is seldom relevant and the overhead volume groups employ is therefore not needed. It is better to assign a PVID by simply changing the attribute of each individual hdisk. For instance, the VIOS command: `chdev -dev hdisk03 -attr pv=yes` will assign a PVID to hdisk3.

2. For VIOS disks, ***use available location information to aid your RAID planning***. To obtain RAID sets in IBM i operating system, you simply point DST at particular groups you want and IBM i operating system decides which disks go together. Under VIOS, for internal disks, you have to do this yourself. The names help show you what to do. For instance, suppose VIOS shows the following for a set of internal disks:

```
Name      Location      State       Description       Size
pdisk0   07-08-00-2,0  Active      Array Member      35.1GB
pdisk1   07-08-00-3,0  Active      Array Member      35.1GB
pdisk2   07-08-00-4,0  Active      Array Member      35.1GB
pdisk3   07-08-00-5,0  Active      Array Member      35.1GB
pdisk4   07-08-00-6,0  Active      Array Member      35.1GB
pdisk5   07-08-01-0,0  Active      Array Member      35.1GB
pdisk6   07-08-01-1,0  Active      Array Member      35.1GB
```

Here, it turns out that these particular physical disks are on two internal SCSI buses (00 and 01) and have device IDs of 2, 3, 4, 5, and 6 on SCSI bus 00 and device IDs of 0 and 1 on SCSI bus 01. If this was all there was, a three disk RAID set of pdisk0, pdisk1, and pdisk5 would be a good choice. Why? Because pdisk0 and 1 are on internal SCSI bus 00 and the other one is one SCSI bus 01. That provides a good balance for the available drives. This could also be repeated for pdisk2, pdisk3, pdisk4, and pdisk6. This would result in two virtual drives being created to represent the seven physical drives. The fact that these are two RAID5 disk sets (of three and four physical disks, respectively) would be unknown to IBM i operating system, but managed instead by VIOS. One or may be two virtual SCSI buses would be required to present them to IBM i operating system by VIOS. A large configuration could provide for RAID5 balance over even more SCSI buses (real and virtual).

On external storage, the discussion is slightly more complicated, because these products tend to package data into LUNs that already involve multiple physical drives. Your RAID set work would have to use whatever the external disk storage product gives you to work with in terms of naming conventions and what degree of control you have available to reflect favorable physical boundaries. Still, the principles are the same.

3. ***Limited number of virtual devices per virtual SCSI adapter.*** You will have to configure some number of virtual SCSI adapters so that VIOS can provide a path for IBM i operating system to talk to VIOS as if these were really physical SCSI devices. These adapters, in turn, implement some existing rules, so that only 16 virtual disks can be made part of a given virtual adapter. You probably would not want to exceed this limit anyway. Note that the virtual adapters need not relate to physical boundaries of the various underlying devices. The main issue is to balance the load. You may be able to segregate data base and journal data at this level. Note also that in a proper configuration, the virtual SCSI adapters will carry command traffic only. The actual data DMA will be direct to the IBM i operating system partition.

4. ***VIOS and Shared Processors.*** On the whole, dedicated VIOS processors will work better than shared processors, especially as the IBM i operating system partition needs three or more CPUs itself. If you do not need shared processors for other reasons, experiment and see if dedicated VIOS processors work better. In fact, it might be an experiment worth running even if you have shared processors configured generally.

5. ***VIOS and memory.*** VIOS arranges for the DMA to go directly to the IBM i operating system memory (with the help of PHYP and IBM i operating system to ensure integrity). This means that actual data transfer will not go through VIOS. It only needs enough main storage to deal with managing disk traffic, not the data the traffic itself consumes. Our current measurements suggests that 1 GB of main storage is the minimum recommended. Other work suggest that unless substantial virtual LAN is involved, between 1 GB and 2 GB tends to suffice at the 1 to 3 CPU ratio we typically measured.

6. ***VIOS and Queue Depth***. Queue depth is a value you can change, so one can experiment to find the best value, at least on a per IPL basis. VIOS tends the set the queue depth parameter to smaller values. Especially if you follow our recommendations for the number of virtual disks, you will find values like 32 to work well for the device as a starting point. If you do that, you will also want to set the queue depth for the adapter (usually called num_cmd_elems) to its larger value, often 512. Consult the documentation.

### 14.5.1.3 VIOS and JS12 Express and JS22 Express Considerations

Most of our work consisted of measurements with the JS22 offering and external disks using the DS4800 product. The following are results obtained in various measurements and then a few general comments about configuration will follow.

#### 14.5.1.3.1 BladeCenter H JS22 Express running IBM i operating system/VIOS

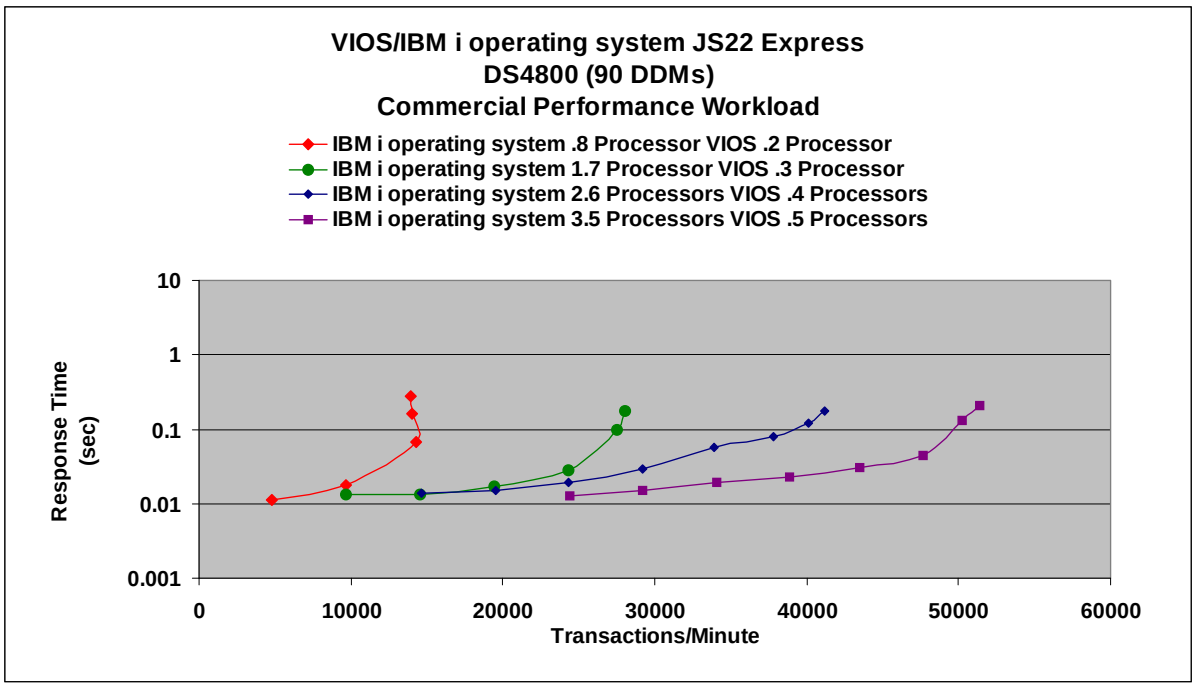
The following tests were run using a 4 processor JS22 Express in a BladeCenter H chassis, 32 GB of memory and a DS4800 with a total of 90 DDMs, (8 DDMs using RAID1 externalized in 2 LUNs for the system ASP, 6 DDMs in each of 12 RAID1 LUNs (a total of 72 DDMs) in the database ASP, and 10 DDMs unprotected externalized in 2 LUNs for the journal ASP). We had two Fibre Channel attachments to the DS4800 with half of the LUNs in each of the ASP's using controller A as the preferred path and the other half of the LUNs using controller B as the preferred path. The following charts show some of the performance characteristics we observed running our Commercial Performance Workload in our test environment. Your results may vary based on the characteristics of your workload. A description of the Commercial Performance Workload can be found in appendix A of the Performance Capabilities Reference.

Creating and running multiple LPARs can lead to unique system management challenges. Reference. The following is a link to an LPAR white paper.
http://www.ibm.com/systems/i/solutions/perfmgmt/pdf/lparperf.pdf. .

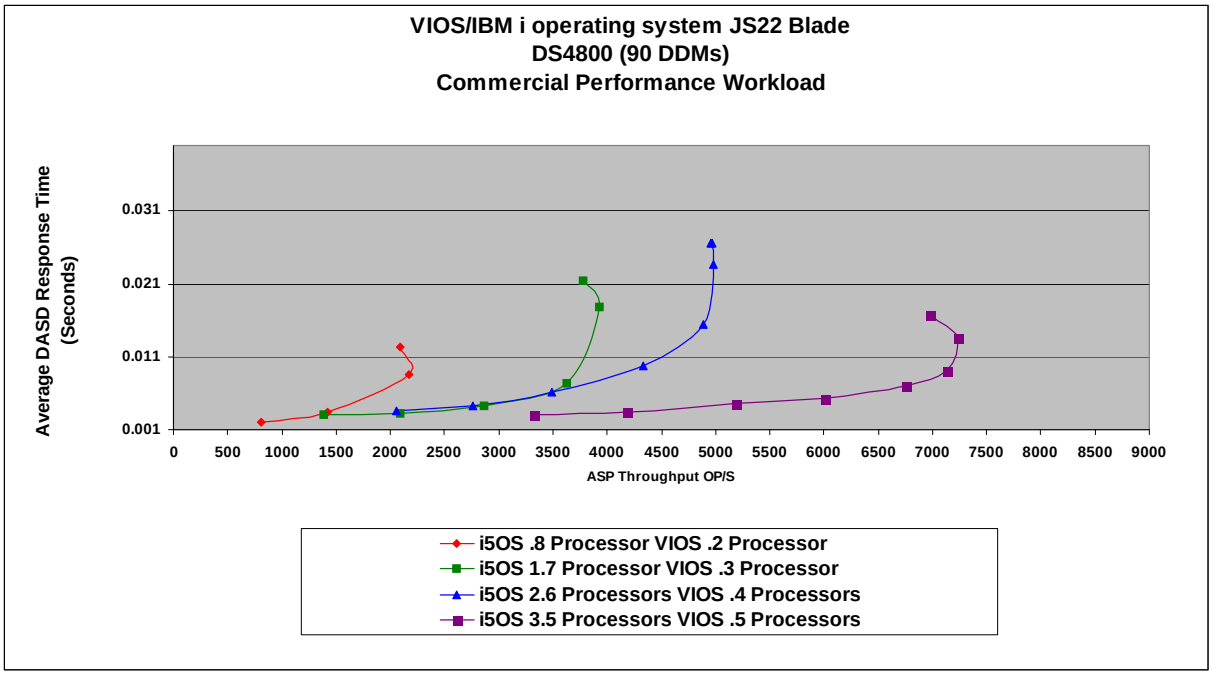
For most of our testing we only utilized one IBM i operating system partition on our JS22 Express. Note that VIOS is the base operating system on the JS22 Express, installed on the internal SAS Disk and VIOS must virtualize the DS4800 LUNs and communication resources to the IBM i operating system partition, which resides on the DS4800 DDMs.

VIOS/IVM must have some of the memory and processor resources for this vrtulization. The amount of resources needed will be dependent on the physical hardware in the Blade Center and the number of partitions being supported on a particular Blade. For our testing we found that we could not operate with under 1 GB of memory and for all of the tests in this section we used 2 GB of memory. The number of processors varied for each experiment and the charts will define the processors used in that experiment.

One important thing to note is that we only changed the amount of memory and processors in the VIOS partition. Otherwise the rest of the settings for the VIOS partition are as they default when the basic configurations is created during the VIOS install. So the VIOS partition processors in my experiments are always set up as shared, only the IBM i operating system partition is created using dedicated processors.



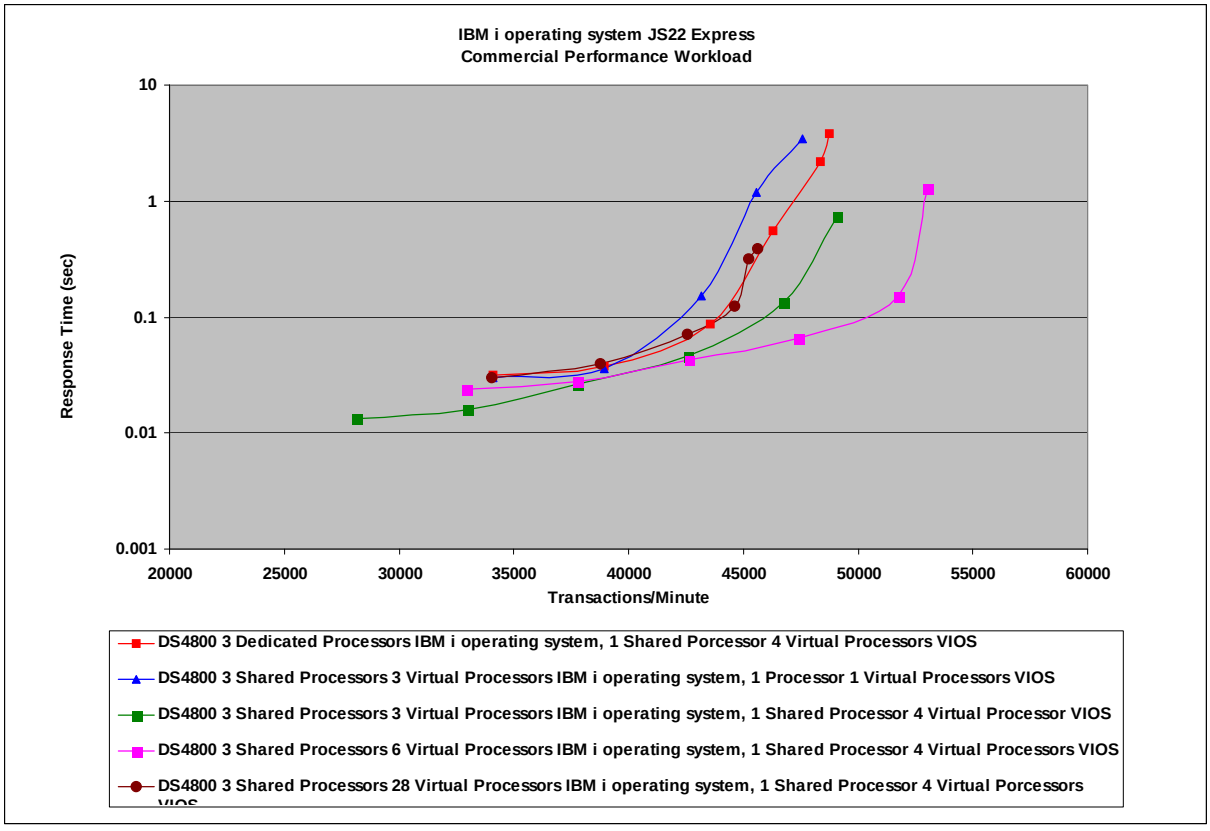

The chart above shows some basic performance scaling for 1, 2, 3 and 4 processors. For this comparison both partition measurements were done with the processors set up as shared, and with the IBM i operating system partition set to capped. The rest of the resources stay constant, which consists of 90 RAID1 DDMs in a DS4800 under 16 LUNs 2 GB of memory assigned to VIOS and 28 GB assigned to the IBM i operating system partition. Note that only 1 LPAR is running at the time of the experiment.



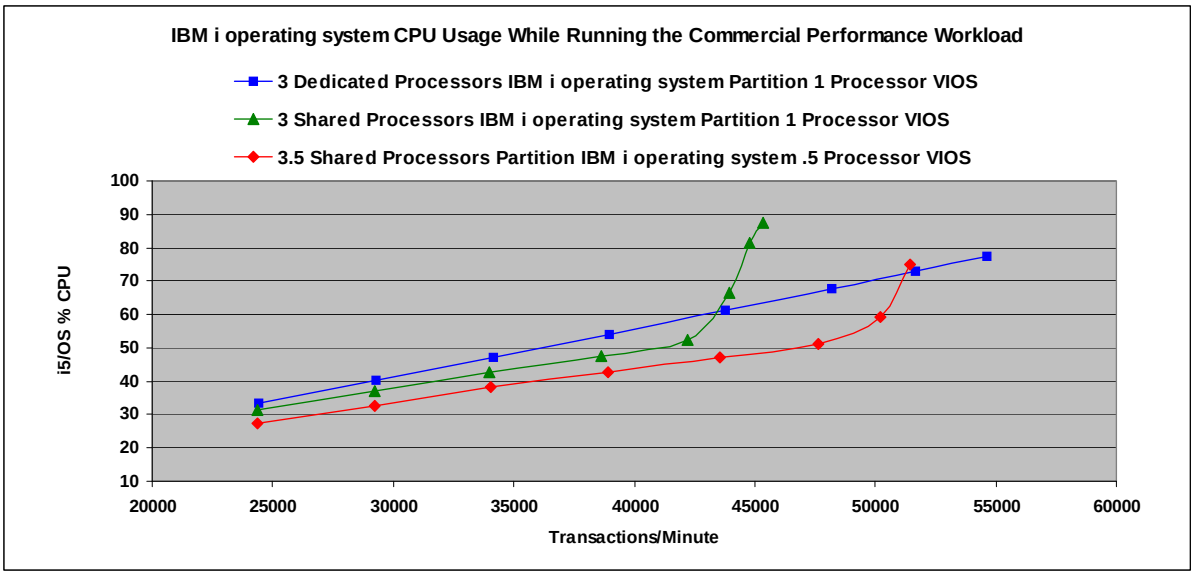
The following charts are a view of the characteristics we observed during our Commercial Performance Workload testing on our JS22 Express. The first chart shows the effect on the Commercial Performance Workload when we apply 3 Dedicated processors and then switch to 3 shared processors. Then incremented the number of virtual processors available.

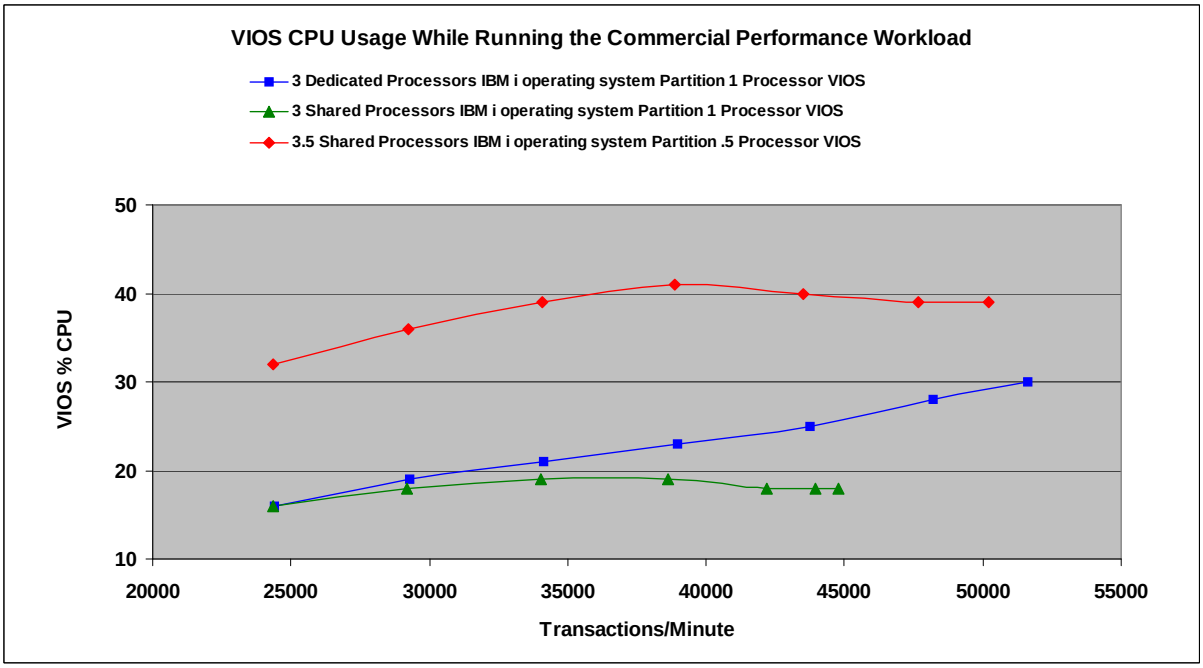
The “red line” is our dedicated processor set up, which is our baseline. The “blue line” is turning on shared processors in what I might have thought of as a fair comparison where 1 virtual processor was assigned for each real processor, resulting in 1 virtual processor for VIOS and 3 virtual processors for IBM i operating system. The Next experiment the “green line” was to increase the number of virtual processors assigned to VIOS but not the number of virtual processors assigned to IBM i operating system. Four virtual processors assigned to VIOS seemed to worked best for our environment. Next was to increase the number of virtual processors assigned to the IBM i operating system environment. Six virtual processors seen in the “purple line” optimized our environment best. As I increased from 6 virtual processors I started losing performance until I had increased to the 28 virtual processors available to me shown in the “dark red line”

Not all workloads will react in the same way but it is important to note that a small change to your configuration can have a large influence on your performance positive and negative. .

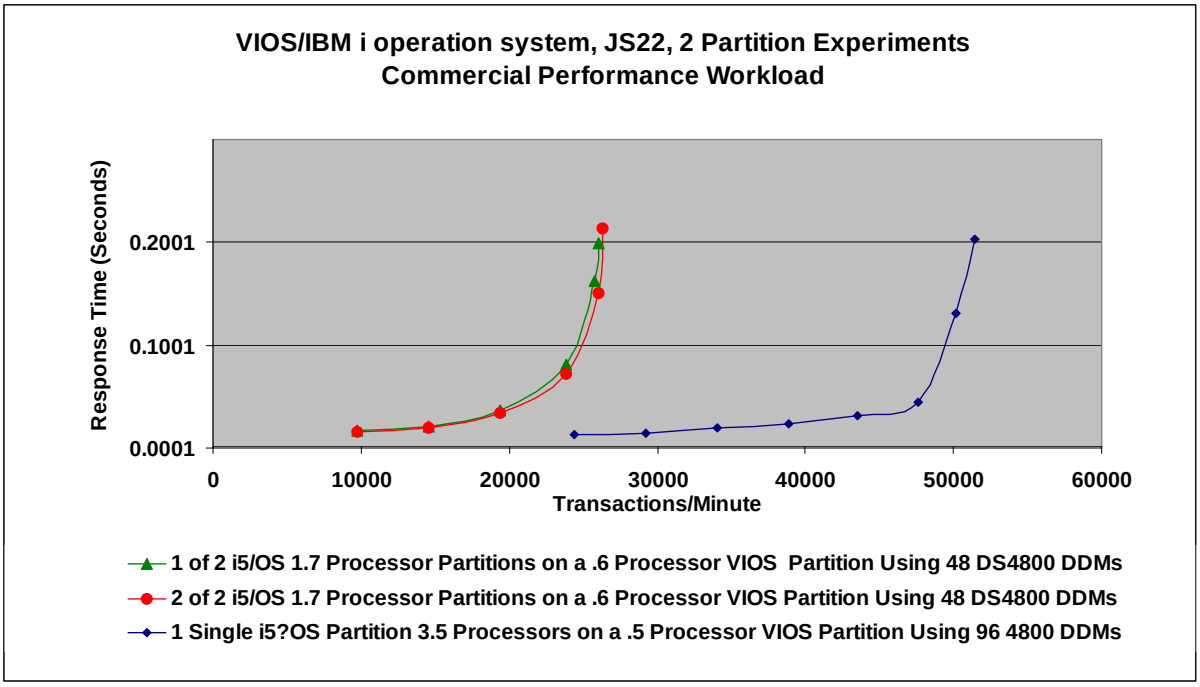


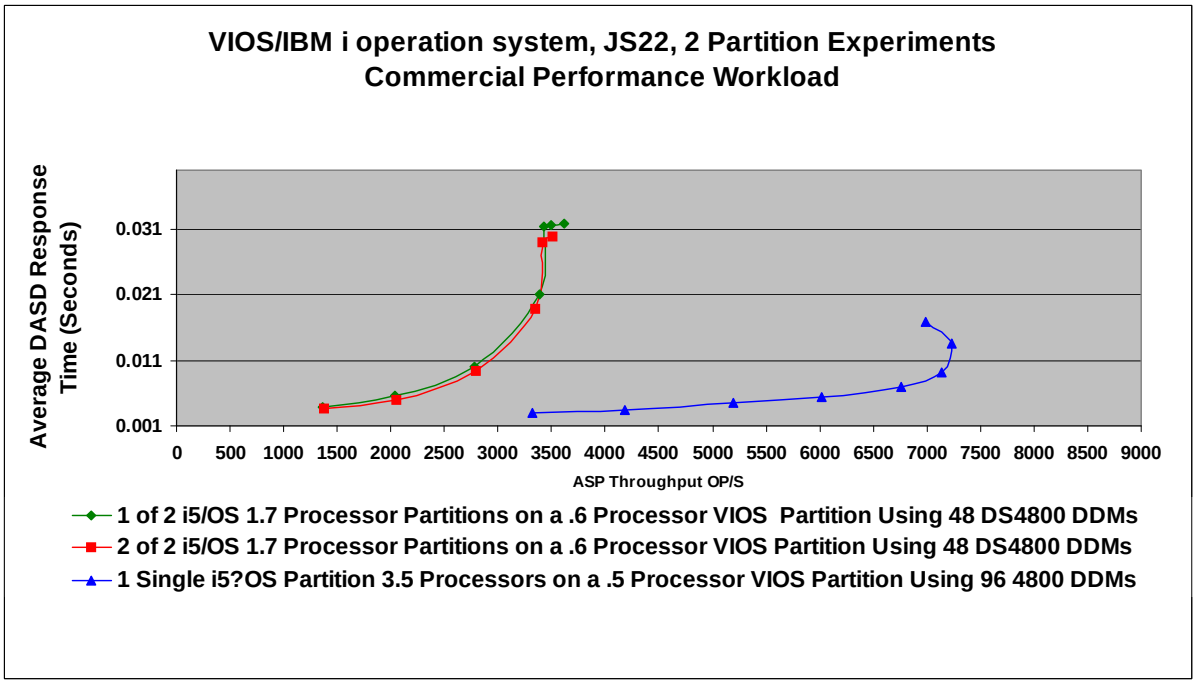
In following single partition Commercial Performance Workload runs the average VIOS CPU stayed under 40%. So we seem to have VIOS resource available but in a lot of customer environments communications and other resources are also running and these resources will also be routed through VIOS.



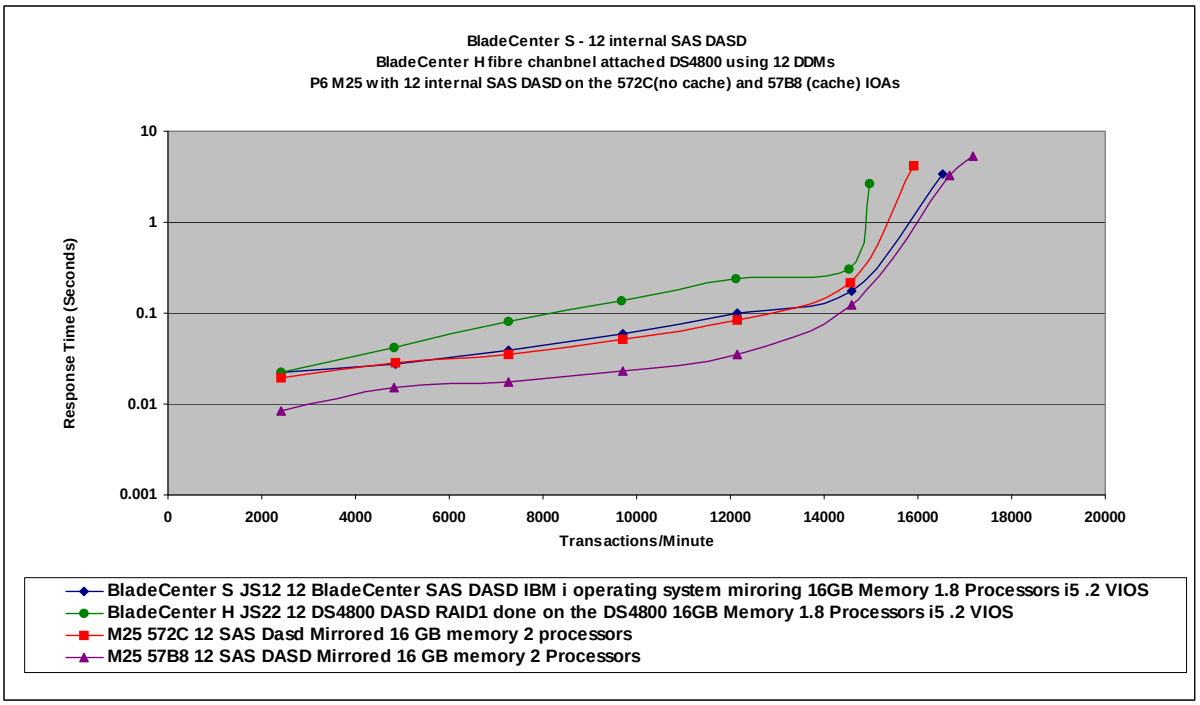
The following chart shows two IBM i operating system partitions using 14GB of memory and 1.7 processors each served by 1 VIOS partition using 2GB of memory and .6 processors. The Commercial Performance Workload was running the same amount of transactions on each of the partitions for the same time intervals. Although there is an observed cost for VIOS to manage multiple partitions, VIOS was able to balance services to the two partitions. Experimenting with the number of processors and memory assigned to the partitions might yield a better environment for other workloads.

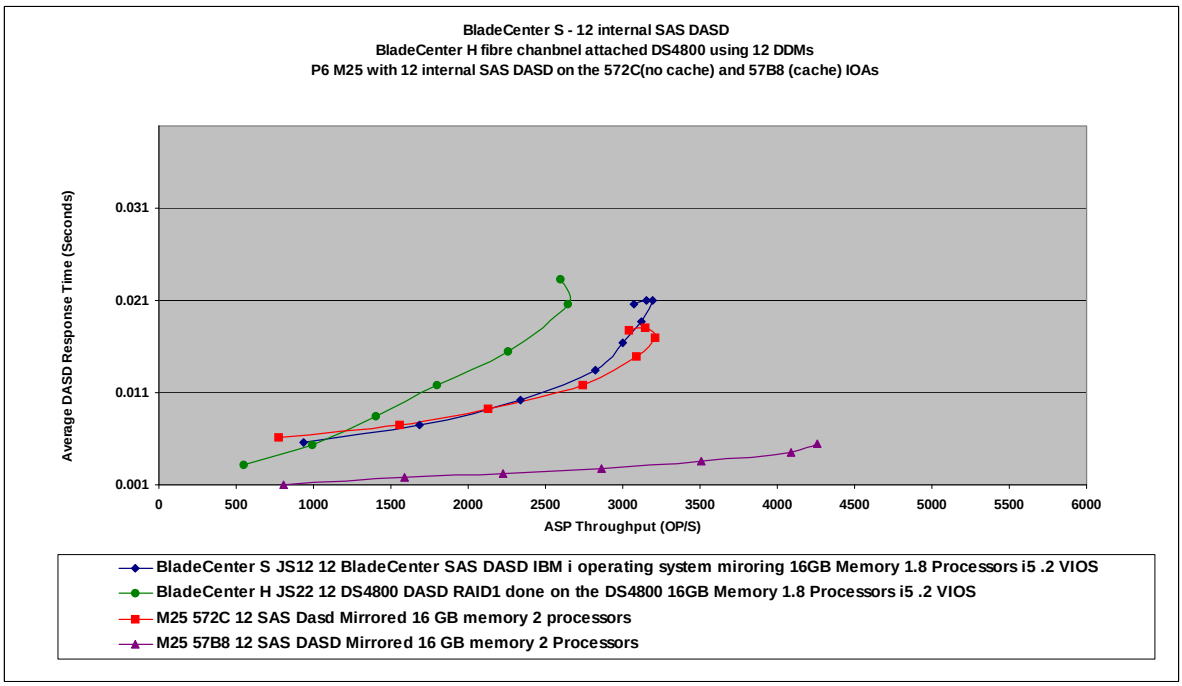


### 14.5.1.3.2 BladeCenter S and JS12 Express

The IBM i operating system is now supported on a JS12 Express in a BladeCenter S. The system is limited to 12 SAS DASD and the following charts try to characterize the performance we achieved during experiments with the Commercial Performance Workload in the IBM lab. Using a JS22 Express in a BladeCenter H connected to a DS4800, we limited the resources in order to get a comparison to the SAS DASD used in the BladeCenter S.



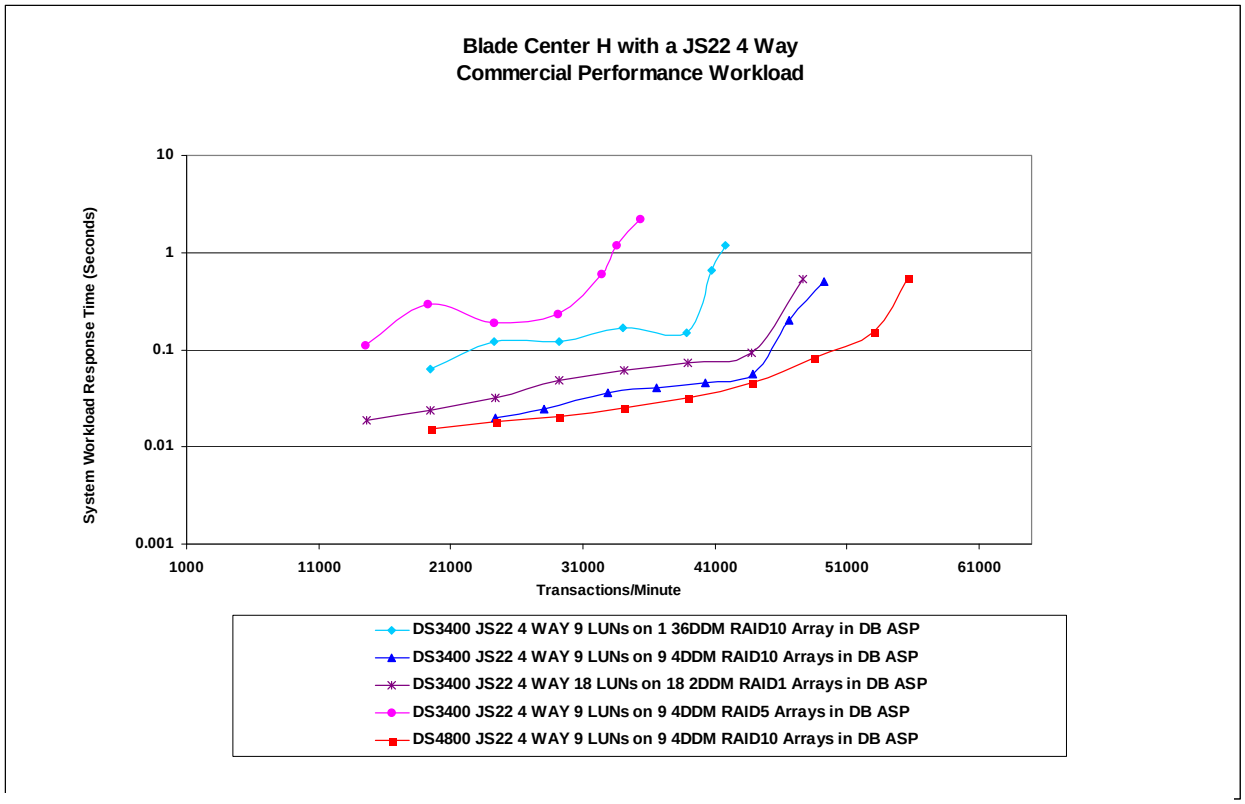
#### 14.5.1.3.3 JS12 Express and JS22 Express Configuration Considerations

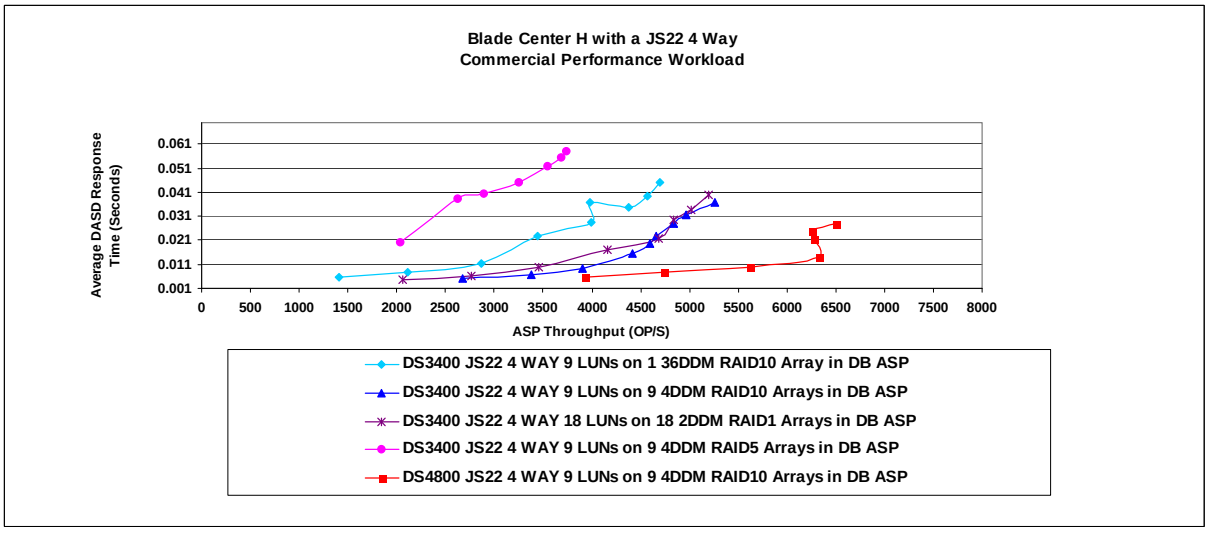
1. The aggregate total of virtual disks (LUNs) will be sixteen at most. Many customers will want to deploy between 12 and 16 LUNs and maximize symmetry. Consult carefully with your support team on the choices here. This is the most important consideration as it is difficult to change later. Consult also any available Best Practices manuals for a given SAN attached storage server.

2. The VIOS partition should be provided with between 1 and 2 GB of memory for disk-based usage's. If virtual LAN is a substantial factor, more memory may be required.

#### 14.5.1.3.4 DS3000/DS4000 Storage Subsystem Performance Tips

**Physical disks can be configured various ways with RAID levels, number of disks in each array and number of LUNs created over those arrays. There are also various reasons for the configurations that are chosen. One end user might be looking for ease of use and choose to create one array with multiple LUNs, where another end user might consider performance to be a more critical issue and select to create multiple arrays. The following charts are meant to show possible performance affects of various configurations using the Commercial Performance Workload.**



## 14.6 IBM i operating system 5.4 Virtual SCSI Performance

The primary goal of virtualization is to lower the total cost of ownership of equipment by improving utilization of the overall system resources and reducing the labor requirements to operate and manage many servers.

With virtualization, the IBM Power Systems can now be used similar to the way mainframes have been used for decades, sharing the hardware between many programs, services, applications, or users. Of course, for each of these individual users of the hardware, sharing resources may result in lower performance than having dedicated hardware, but the overall cost is usually far less than when dedicating hardware to each user. The decision of using virtualization is therefore a trade-off between cost and performance.

IBM i operating system Virtual SCSI is based on a client/server relationship. A IBM i operating system Server partition owns the physical resources, and client partitions access the virtual SCSI resources provided by the IBM i operating system Server partition. The IBM i operating system Server partition has physically attached I/O devices and exports one or more of these devices to other partitions. The client partition is a partition that has a virtual disk and relies on the IBM i operating system Server partition to provide access to one or more physical devices. POWER5 and future POWER technologies provide virtual SCSI support for AIX 5L V5.3 and Linux. Previous POWER technology supported Linux virtual SCSI.

The performance considerations that we detail in this section must be balanced against the savings made on the overall system cost. For example, the smallest physical disk that is available to the IBM i operating system is 70 GB. An AIX or Linux operating system requires only 4 GB of disk. If one disk is dedicated to the operating system, nearly 95% of this physical disk space is unused. Furthermore, the system disk I/O rate is often very low. With the help of IBM i operating system Virtual SCSI, it is possible to split the same disk into 9 virtual disks of about 8 GB each. If each of these disks is used for installation of the operating system, you can support nine separate instances of the operating system, with nine times fewer disks and perhaps as many physical SCSI adapters. Compare these savings with the extra cost of processing power needed to handle the virtual disks.

Enabling IBM i operating system Virtual SCSI results in using extra processing power compared to directly attached disks, due to extra POWER VIO activity. Depending on the configuration, this may or may not yield the same performance when comparing virtual hosted disk devices to physically attached SCSI devices. If a partition has high performance and disk I/O requirements that justify the cost of dedicated hardware, then using virtual SCSI is not recommended. However, partitions with non-critical performance and low disk I/O requirements often can be configured to use virtual SCSI, which in turn lowers hardware and operating costs.

In the test results that follow, we see the CPU required for IBM i operating system Virtual SCSI server and the benefits of the IBM i operating system Virtual SCSI implementation should be assessed for a given environment. Simultaneous multithreading should be enabled in a virtual hosted disk environment. For most efficient virtual hosted disk implementation with larger IO loads, it may be advantageous to keep the IBM i operating system Virtual SCSI Server partition as a dedicated processor. Processor micro partitioning should be used with low IO loads or with workloads which are not latency dependent.

Virtual storage can be created in an ASP using the CRTNWSSTG and linked using the CRTNWSD commands. The disk can be manipulated in the client AIX or Linux partition the same as an ordinary physical disk. Some performance considerations from dedicated storage are still applicable when using virtual storage, such as spreading ASP's across multiple disks on multiple RAID adapters so that parallel access is possible. From the server's point of view, a virtual drive can be served using an entire ASP, or a portion of an ASP. If the server partition provides the client with a partition of a drive, then the server decides the area of the drive to serve to the client when the network storage space is created.
This allows reads and writes of an ASP to be shared among several virtual devices. If the entire ASP is served to the client, then the rules and procedures apply on the client side as if the drive were local

Consider the following general performance issues when using virtual SCSI:

- Only use virtual hosted disk in low I/O loads
- Virtual hosted disk is a client/server model, so the combined CPU cycles required on the I/O client and the I/O server will always be higher than local I/O
- If multiple partitions are competing for resources from a virtual hosted disk server, care must be taken to ensure that enough server resources (processor, memory, and disk) are allocated to do the job.
- There is data read caching in memory on the Virtual hosted disk Server partition. Thus, all I/Os that it services could benefit from effects of caching heavily used methods. Read performance can be improved by increasing the memory in the virtual hosted disk server.

### 14.6.1 Introduction

In general, applications are functionally isolated from the exact nature of their storage subsystems by the operating system. An application does not have to be aware of whether its storage is contained on one type of disk or another when performing I/O. But different I/O subsystems have subtly different performance qualities, and virtual SCSI is no exception. What differences might an application observe using IBM i operating system Virtual SCSI versus directly attached storage? Broadly, we can categorize the possibilities into I/O latency and I/O bandwidth.

We define *I/O response time* as the time that passes between the initiation of I/O and completion as observed by the application. Latency is a very important attribute of disk I/O. Consider a program that performs 1000 random disk I/Os, one at a time. If the time to complete an average I/O is six milliseconds, the application will run no less than 6 seconds. However, if the average I/O response time is reduced to three milliseconds, the application's run time could be reduced by three seconds. Applications that are multi-threaded or use asynchronous I/O may be less sensitive to latency, but under most circumstances, less latency is better for performance.

We define *I/O bandwidth* as the maximum amount of data that can be read or written to storage in a unit of time. Bandwidth can be measured from a single thread or from a set of threads executing concurrently. Though many applications are more sensitive to latency than bandwidth, bandwidth is crucial for many typical operations such as backup and restore of persistent data.

Because disks are mechanical devices, they tend to be rather slow when compared to high-performance microprocessors such as IBM POWER Systems. As such, we will show that virtual hosted disk performance is comparable to directly attached storage under most workload environments.
IBM i operating system hosts disk space in a Network Storage Space (NWSSTG). A network server description (NWSD) is used to give a name to the configuration, to provide an interface for starting and stopping an AIX logical partition, and to provide a link between AIX and its virtual storage.

There are many factors that affect IBM i operating system performance in a virtual SCSI environment. This chapter discusses some of the common factors and offers guidance on how to help achieve the best possible performance. Much of the information in this chapter was obtained as a result of analysis experience within the Rochester development laboratory. Many of the performance claims are based on supporting performance measurement and analysis with a primitive disk workload. In some cases, the actual performance data is included here to reinforce the performance claims and to demonstrate capacity characteristics.

All measurements were completed on a POWER5 570+ 4-Way (2.2 GHz). Each system is configured as an LPAR, and each virtual SCSI test was performed between two partitions on the same system with one CPU for each partition. IBM i operating system 5.4 was used on the virtual SCSI server and AIX 5.3 was used on the client partitions.

The primitive disk workload used to evaluative the performance of virtual SCSI is an in house, multi-processed application that performs all types of Synchronous or Asynchronous I/O (read/write/sequential/random) to a target device. The program is run on an AIX or Linux client and gets reports of CPU consumption and gathers disk statistics. Remote statistics are gathered via a socket based application which gathers CPU from the IBM i operating system hosted disk and physical disk statistics.

The purpose of this document is to help virtual SCSI users to better understand the performance of their virtual SCSI system. A customer should be able to size the expected speed of their application from this document.

**Note:** You will see different terms in this publication that refer to the various components involved with virtual SCSI. Depending on the context, these terms may vary. With SCSI, usually the terms server and client are used, so you may see terms such as virtual SCSI client and virtual SCSI server. On the Hardware Management Console, the terms virtual SCSI server adapter and virtual SCSI client adapter are used. They refer to the same thing. When describing the client/server relationship between the partitions involved in virtual SCSI, the terms hosting partition (meaning the IBM i operating system Server) and hosted partition (meaning the client partition) are used.

---

### 14.6.2 Virtual SCSI Performance Examples

The following sections compare virtual to native I/O performance on bandwidth tests. In these tests, a single thread operates sequentially on a constant file that is 6GB in size, with a dedicated IBM i operating system Server partition. More I/O operations are issued when reading or writing to the file using a small block size than with a larger block size. Because of the larger number of operations and the fact that each operation has a fixed amount of overhead regardless of transfer length, the bandwidth measured with small block sizes is much lower than with large block sizes.

For tests with multiple Network Storage Spaces (NWSS), a thread operates sequentially for each network storage space on a constant file that is 6GB in size, again with a dedicated IBM i operating system Server partition. The following sections compare native vs. virtual, multiple network storage spaces, multiple network storage descriptions, and disk scaling.

### 14.6.2.1 Native vs. Virtual Performance

Figure 1 shows a comparison of measured bandwidth using virtual SCSI and local attached DASD for reads with varying block sizes of operations. The difference in the reads between virtual I/O and native I/O in these tests is attributable to the increased latency using virtual I/O. The difference in writes is caused by misalignment, which causes a read for every write. A write alignment change is planned for a future IBM i operating system release which will make virtual and native writes similar in speed.

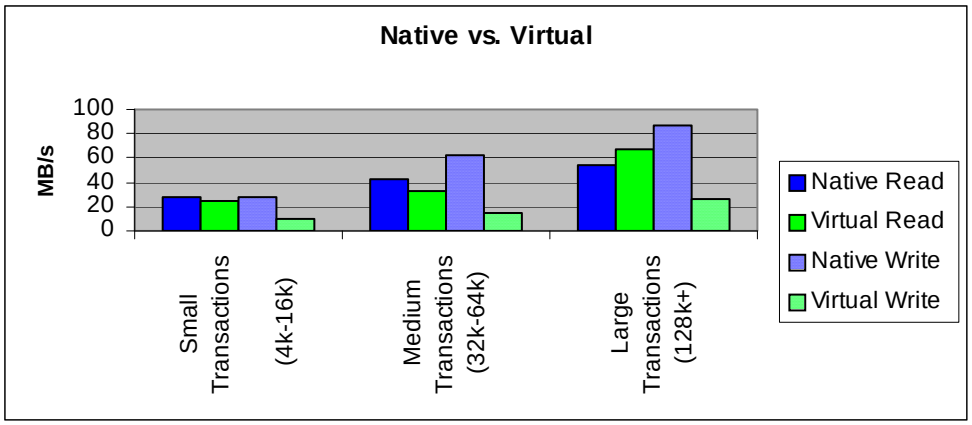
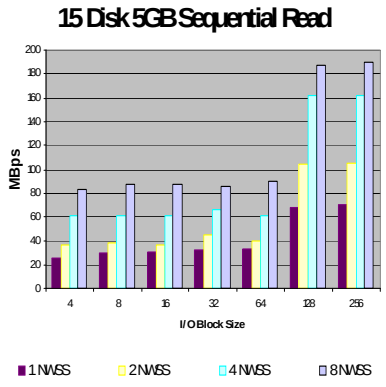



*Figure 1 - The figure above shows a comparison of native vs virtual. This experiment shows that virtual write performance is significantly less then native. Read performance performs similar or better then native depending on the read-cache performance.*

### 14.6.2.2 Virtual SCSI Bandwidth-Multiple Network Storage Spaces

Figure 2 shows a comparison of measured bandwidth while scaling network storage spaces with varying block sizes of operations. The difference in the scaling of these tests is attributable to the performance gain, which can be achieved by adding multiple network storage spaces. This experiment shows that in order to achieve better performance from the hard disk, multiple network storage spaces can be used.



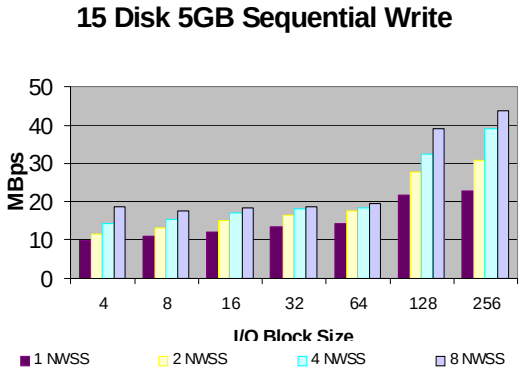




*Figure 2- The figures above show performance while scaling network storage spaces. This experiment shows that adding NWSS increases the throughput for read/write performance. The best performance is achieved by using 8 NWSS.*

### 14.6.2.3 Virtual SCSI Bandwidth-Network Storage Description (NWSD) Scaling

Figure 3 shows a comparison of measured bandwidth while scaling network storage descriptions with varying block sizes of operations. Each of the network storage descriptions have a single network storage space attached to them. The difference in the scaling of these tests is attributable to the performance gain which can be achieved by adding multiple network storage descriptions. This experiment shows that in order to achieve better write performance from the hard disk, multiple network storage descriptions can be used. In order to achieve better performance, 1 network storage space should be used for every 2-4 disk drives in the ASP and each network storage space should be attached to its own network storage description.

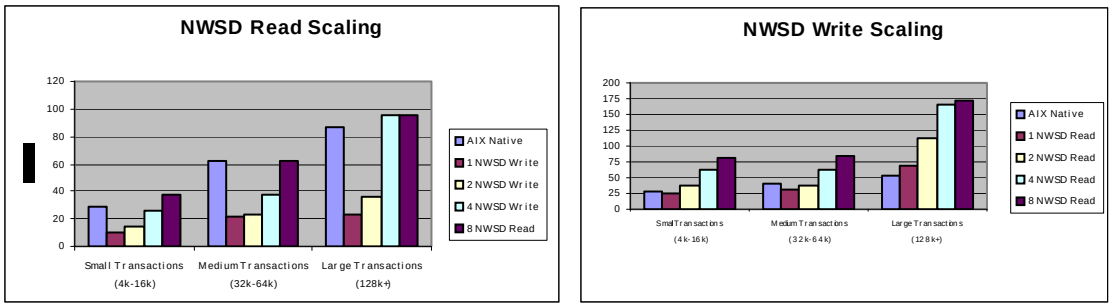



*Figure 3 The figures above show performance while scaling network storage descriptions. This experiment shows that adding NWSD increases the throughput for write performance, which was not achievable using 1 network storage description. Read performance increases similar to the network storage space scaling figure.*

### 14.6.2.4 Virtual SCSI Bandwidth-Disk Scaling

Figure 4 shows a comparison of measured bandwidth while scaling disk drives with varying block sizes of operations. Each of the network storage descriptions have a single network storage space attached to them. The difference in the scaling of these tests is attributable to the performance gain which can be achieved by adding disk drives and IO adapters. The figures below include small (4k-64k) transactions and larger (128k) transactions.

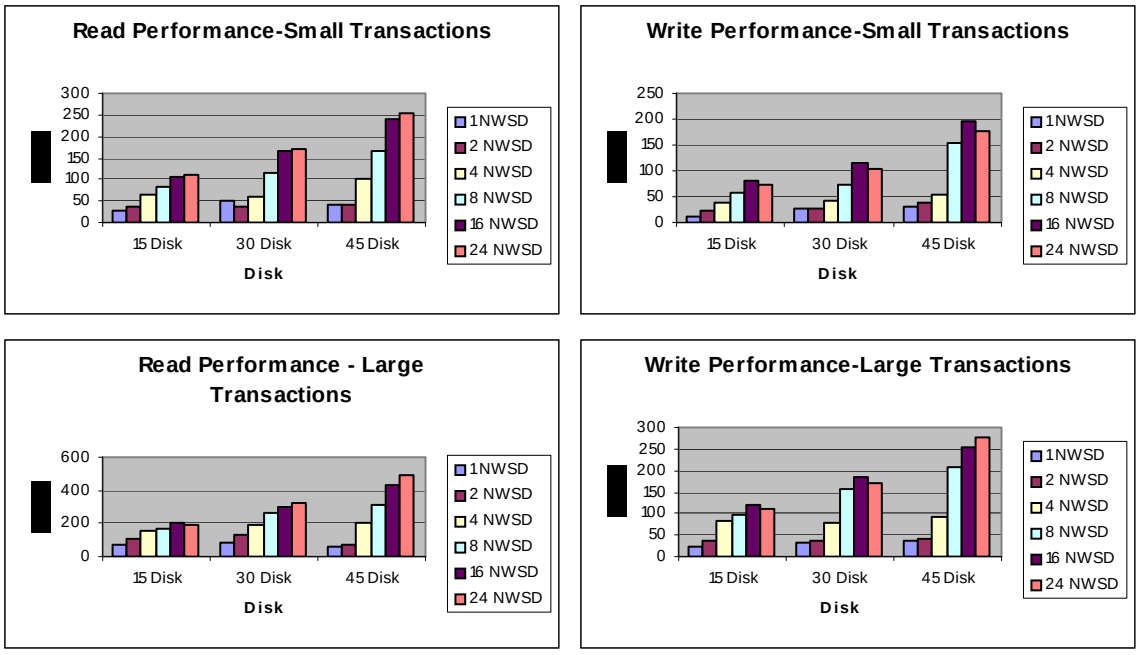



*Figure 4 The figures above show read and write performance for small (4k-64k) and large transactions (128k+). This experiment shows that adding disk drives increases the throughput. A system with 45 disk drives will be able to transfer approximately 3 times faster then a system with 15 disk drives. Notice 24-network storage descriptions were used in order to achieve maximum performance.*

### 14.6.3 Sizing

Sizing methodology is based on the observation that processor time required to perform an I/O on the IBM i operating system Virtual SCSI server is fairly constant for a given I/O size. The I/O devices supported by the Virtual SCSI server are sufficiently similar to provide good recommendations. These numbers are measured at the physical processor.

There are considerations to address when designing and implementing a Virtual SCSI environment. The primary considerations are:

- Dedicated processor server partitions or Micro-Partitioning
- Server partition memory requirements
- One thing that does not have to be factored into sizing is the processor impact of using Virtual I/O on the client. The processor cycles executed on the client to perform a Virtual SCSI I/O are comparable to that of a locally attached I/O. Thus, there is no increase or decrease in sizing on the client partition for a known task.

#### 14.6.3.1 Sizing when using Dedicated Processors

One sizing method is to size the Virtual SCSI server to the maximum I/O rate of the attached storage subsystem. The sizing could be biased to small I/Os or large I/Os. Sizing to maximum capacity for large I/Os balances the processor capacity of the Virtual SCSI server to the potential I/O bandwidth of the attached I/O. The negative facet of this sizing methodology is that, in nearly every case, we will assign more processor entitlement to the Virtual SCSI server than it typically consumes.

Consider a case where an I/O server manages 15 physical SCSI disks. We can arrive at an upper bound of processors required based on assumptions about the I/O rates that the disks can achieve. If it is known that the workload is dominated by 16 KB operations, we could assume that the 15 disks are capable of 1 read transaction every 36 milliseconds. An IBM i operating system Virtual SCSI server could support around 30,000 read transactions per second on a single processor provided enough disk were present.

To calculate IBM i operating system Virtual SCSI CPU requirements the following formula is provided. The number of transactions per second could be collected by the IBM i operating system command WRKDSKSTS. Based on the average transaction size in WRKDSKSTS, select a number from the table.

| | | Size of IO | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 4 | 8 | 16 | 32 | 64 | 128 | 256 |
| Type of Transaction | Read | 16 | 22 | 34 | 57 | 92 | 163 | 314 |
| | Write | 21 | 26 | 36 | 54 | 82 | 148 | 282 |

*Figure 5- CPU milliseconds to process virtual SCSI I/O transaction*

The table above shows the time in milliseconds per transaction that Virtual SCSI takes to process one transaction. This value can be used in the formula below to estimate the amount of CPU required per a partition.

$$\frac{(\#\ \text{of Transactions per second} * \text{Time in Milliseconds per transaction})}{1{,}000{,}000} = \text{CPU Utilization}$$

For example.. If your workload performed 10,000 16k read transactions the equation would look like this (34 was selected from the table above):

$$\frac{(10{,}000 * 34)}{1{,}000{,}000} = .34 \text{(34\% of a total CPU)}$$

The total CPU required for a workload, which performs 10,000 16k read transactions per second, would be 34% of a 2.2Ghz POWER5 processor. If a different size processor is used adjust these numbers accordingly. Remember the number chosen in WRKDSKSTS is an average of all I/O's. Your workload could be a mixture of very large transactions and very small transactions. This is to provide a guideline of how to size your CPU correctly, and your results might vary.

Using Dedicated processor partitions may require more CPU then necessary that could be used by other partitions, but will guarantee peak performance. It is most effective if the average I/O size can be estimated so that peak bandwidth does not have to be assumed. Most Virtual SCSI servers will not run at maximum I/O rates all the time, so the use of surplus processor time is potentially wasted by using dedicated processor partitions.

### 14.6.3.2 Sizing when using Micro-Partitioning

Defining Virtual SCSI servers in micro-partitions enables much better granularity of processor resource sizing and potential recovery of unused processor time by uncapped partitions. Tempering those benefits, use of micro-partitions for Virtual SCSI servers slightly increases I/O response time and creates somewhat more complex processor entitlement sizing.

The sizing methodology should be based on the same operation costs as for IBM i operating system Server partitions. However, additional entitlement should be added for running in micro-partitions. We recommend that the IBM i operating system Server partition be configured as uncapped so it can take advantage of unused capacity of other partitions, it is possible to get more processor time to service I/O.

Because I/O latency with Virtual SCSI varies with the machine utilization and IBM i operating system Server topology, consider the following:

1. For the most demanding I/O traffic (high bandwidth or very low latency), try to use native I/O.
2. If using Virtual I/O and the system contains enough processors, consider putting the IBM i operating system Server in a dedicated processor partition.
3. If using a Micro-Partitioning IBM i operating system Server, use as few virtual processors as possible.
4. In order to avoid latency issues try to always size the CPU generously

### 14.6.3.3 Sizing memory

The IBM i operating system Virtual SCSI server supports data read caching on the virtual hosted disk server partition. Thus all I/Os that it services could benefit from effects of caching heavily used data. Read performance can vary depending upon the amount of memory which is assigned to the server partition. Workloads which have a small memory footprint can improve their performance greatly by increasing the amount of memory in the IBM i operating system Virtual SCSI server. Alternatively, a system which works on a large amount of data may not see any benefit from caching. The memory for the IBM i operating system Virtual SCSI server in this case can be set at less then 1 GB.

One method to size this is to begin by looking at your ASP in which your network storage space is located. While the system is running the desired workload, type in the command WRKDSKSTS. Write down the average number of I/O request per second in the ASP which is being used by the network storage space. Now dynamically add memory to the partition. Check the number of I/O requests per second once again (remember to reset the statistics using F10). The number of I/O requests per second should lower and your throughput to the IBM i operating system Virtual SCSI server should increase.

Continue adding memory to the IBM i operating system server until you no longer see the number of I/O requests per second change. If your workload changes at a later date the memory can be readjusted accordingly.

Figure 6 below shows a comparison of measured bandwidth of cached transactions with varying block sizes of operations. The figure includes small (4k-64k) transactions and larger(128k) transactions. A partition which runs completely from memory can experience throughput rates as high as 6GB/sec. If it is memory constrained the systems throughput will be lower.

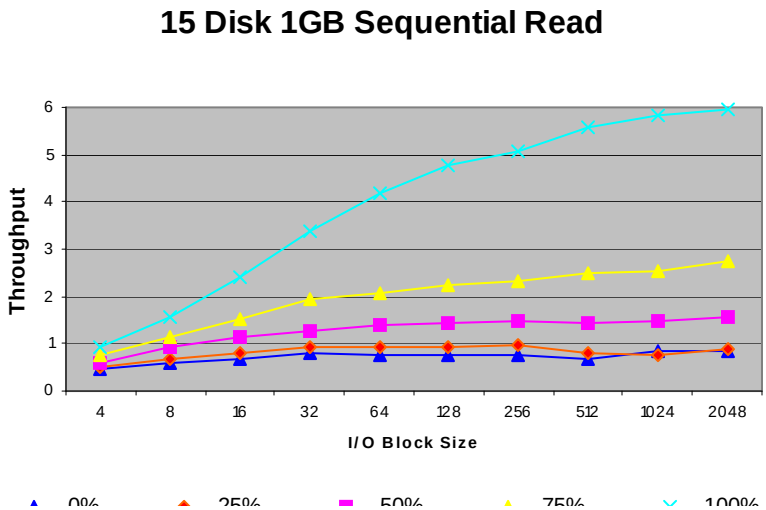



Figure 6 - *The figure above shows a comparison of small, medium, and large transactions affect on memory if cached. The lines represent the amount of data, which is cached in memory. The efficiency of I/O improves with cache hits and larger I/O size. Effectively, there is a fixed latency to start and complete an I/O, with some additional cycle time based on the size of the I/O.*

### 14.6.4 AIX Virtual IO Client Performance Guide

The following is a link which will direct you to more in-depth performance tuning for AIX virtual SCSI client.

Advanced POWER Virtualization on IBM @server p5 Servers: Architecture and Performance Considerations http://www.redbooks.ibm.com/abstracts/sg247940.html?

### 14.6.5 Performance Observations and Tips

- In order to achieve best performance 1 network storage description should be used for every 2-4 disks within an ASP.
- A method to improve write performance is to create 8 NWSD for every 15 disks.
- Best performance was obtained with a network storage description for every network storage space.
- Sizing your memory correctly can improve read performance vastly.
- Multiple network storage descriptions (NWSD) can be attached to a single ASP. No performance benefit from using multiple ASP's was seen.
- For maximum logical volume throughput use multiple network storage spaces attached to a single logical volume.
- With low I/O loads and a small number of partitions, Micro-Partitioning of the IBM i operating system Server partition has little effect on performance.
- For a more efficient Virtual SCSI implementation with larger loads, it may be advantageous to keep the I/O server as a dedicated processor.
- Extensive information can be found at the System i Information Center web site at: http://publib.boulder.ibm.com/iseries.

### 14.6.6 Summary

Virtualization is an innovative technology that redefines the utilization and economics of managing an on demand operating environment. POWER5 and future POWER architectures provide new opportunities for clients to take advantage of virtualization capabilities. IBM i operating system family provides the capability for a single physical I/O adapter to be used by multiple logical partitions of the same server, enabling consolidation of I/O resources.

The system resource cost of Virtual SCSI implementation is small, and clients should assess the benefits of the Virtual SCSI implementation for their environment. Simultaneous multithreading should be enabled in a virtual SCSI environment.

Virtual SCSI implementation is an excellent solution for clients looking to consolidate I/O resources with a modest amount of processor. The new IBM i operating system POWER Systems Virtual SCSI capability creates new opportunities for consolidation, and demonstrates strong performance and manageability.

# Chapter 15. Save/Restore Performance

This chapter's focus is on the **IBM i operating system platform.** For legacy system models, older device attachment cards, and the lower performing backup devices see the V5R3 performance capabilities reference.

Many factors influence the observable performance of save and restore operations. These factors include:

- The backup device models, number of DASD units the data is spread across, processors, LPAR configurations, IOA used to attach the devices.
- Workload type: Large Database File, User Mix, Source File, integrated file system (Domino, Network Storage, 1 Directory Many Objects, Many Directory Many Objects.
- The use of data compression, data compaction, and Optimum Block Size (USEOPTBLK)
- Directory structure can have a dramatic effect on save and restore operations.

## 15.1 Supported Backup Device Rates

As you look at backup devices and their performance rates, you need to understand the backup device hardware and the capabilities of that hardware. The different backup devices and IOAs have different capabilities for handling data for the best results in their target market. The following table contains backup devices and rates. Later in this document the rates are used to help determine possible performance. A study of some customer data showed that compaction on their database file data occurred at a ratio of approximately 2.8 to 1. The database files used for the performance workloads were created to simulate that result.

*Table 15.1.1* backup device speed and compaction information

| Backup Device | Rate (MB/S) | COMPACTION FACTOR |
|---|---|---|
| DVD-RAM | 0.75 Write/2.8 Read | 2.8 #1 |
| SAS DVD-RAM | 2.5 | 2.8 #1 |
| SLR60 | 4.0 | 2.0 |
| SLR100 | 5.0 | 2.0 |
| VXA-2 | 6.0 | 2.0 |
| 6279 VXA-320 | 12.0 | 2.0 |
| 6258 4MM tape Drive | 6.0 | 2.0 |
| 5755 ½ High Ultrium-2 | 18.0 | 2.8 |
| 3580 Ultrium 2 | 35.0 | 2.8 |
| 3592J Fiber Channel | 40.0 | 2.8 |
| 3580 Ultrium 3 Fiber Channel) | 80.0 | 2.0 |
| 5746 Half High Ultrium 4 | 120.0 | 2.0 |
| 3580 Ultrium 4 Fiber Channel | 120.0 | 2.0 |
| 3592E Fiber Channel | 100.0 | 2.5 |

#1. Software compression is used here because the hardware doesn't support device compaction
Note the compaction factor is a number that used with the formulas in the following chapter to help describe the actual rates observed as the lab workloads were run using the above drives. This is not the compression ratio of the data being written to tape. I list them here to help understand what our experiments were able to achieve relative to the published drive speed.

## 15.2 Save Command Parameters that Affect Performance

**Use Optimum Block Size (USEOPTBLK)**
The USEOPTBLK parameter is used to send a larger block of data to backup devices that can take advantage of the larger block size. Every block of data that is sent has a certain amount of overhead that goes with it. This overhead includes block transfer time, IOA overhead, and backup device overhead. The block size does not change the IOA overhead and backup device overhead, but the number of blocks does. For example, sending 8 small blocks will result in 8 times as much IOA overhead and backup device overhead. This allows the actual transfer time of the data to become the gating factor. In this example, 8 software operations with 8 hardware operations essentially become 1 software operation with 1 hardware operation when USEOPTBLK(*YES) is specified. The usual results are significantly lower CPU utilization and the backup device will perform more efficiently.

**Data Compression (DTACPR)**
Data compression is the ability to compress strings of identical characters and mark the beginning of the compressed string with a control byte. Strings of blanks from 2 to 63 bytes are compressed to a single byte. Strings of identical characters between 3 and 63 bytes are compressed to 2 bytes. If a string cannot be compressed a control character is still added which will actually expand the data. This parameter is usually used to conserve storage media. If the backup device does not support data compaction, the system i software can be used to compress the data. This situation can require a considerable amount of CPU.

**Data Compaction (COMPACT)**
Data compaction is the same concept as software compression but available at the hardware level. If you wish to use data compaction, the backup device you choose must support it.

## 15.3 Workloads

The following workloads were designed to help evaluate the performance of single, concurrent and parallel save and restore operations for selected devices. Familiarization with these workloads can help in understanding differences in the save and restore rates.

**Database File related Workloads**:
The following workloads are designed to show some possible customer environments using database files.

**User Mix** **User Mix 3GB, User Mix 12GB** - The User Mix data is contained in a single library and made up of a combination of source files, database files, programs, command objects, data areas, menus, query definitions, etc. User Mix 12GB contains 49,500 objects and User Mix 3GB contains 12,300 objects.

**Source File** **Source File 1GB** - 96 source files with approximately 30,000 members.

**Large Database File** **Large File 4GB, 32GB, 64GB**, **320GB** - The Large Database File workload is a single database file. The members in the 4GB and 32GB files are 4GB in size. The Members in the 64GB and 320GB files are 64GB in size.

**Integrated File System related Workloads:**
Analysis of customer systems indicates about 1.5 to 1 compaction on the tape drives with integrated file system data. This is partly due to the fact that the IBM i operating system programs that store data in the integrated files system, do some disk management functions where they keep the IFS space cleaned up and compressed. And the fact that the objects tend to be smaller by nature, or are mail documents, HTML files or graphic objects that don't compact. The following workloads (1 Directory Many Objects, Many Directories Many Objects, Domino, Network Storage Space) show some possible customer integrated file system environments.

**1 Directory Many objects** This integrated file system workload consists of 111,111 stream files in a single directory where the stream files have 32K of allocated space, 24K of which is data. Approximately 4 GB total sampling size.

**Many Directories Many objects** This integrated file system workload is 6 levels deep, 10 directories wide where each directory level contains 10 directories resulting in a total of 111,111 Directories and 111,111 stream files, where the stream files have 32K of allocated space, 24K of which is data. Approximately 5 GB total sampling size.

**Domino** This integrated file system workload consists of a single directory containing 90 mail files. Each mail file is 152 MB in size. The mail files contain mail documents with attachments where approximately 75% of the 152 MB is attachments. Approximately 13 GB total sampling size.

**Network Storage Space** This integrated file system workload consists of a Linux storage space of approximately 6 GB total sampling size.

## 15.4 Comparing Performance Data

When comparing the performance data in this document with the actual performance on your system, remember that the performance of save and restore operations is data dependent. If the same backup device was used on data from three different systems, three different rates may result.

The performance of save and restore operations are also dependent on the system configuration, most directly affected by the number and type of DASD units on which the data is stored and by the type of storage IOAs being used.

Generally speaking, the Large Database File data that was used in testing for this document was designed to compact at an approximate 2.8 to1 ratio. If we were to write a formula to illustrate how performance ratings are obtained, it would be as follows:

((DeviceSpeed * LossFromWorkLoadType) * Compaction) = MB/Sec * 3600 = MB/HR /1000 = GB/HR.

But the reality of this formula is that the "LossFromWorkLoadType" is far more complex than described here. The different workloads have different overheads, different compaction rates, and the backup devices use different buffer sizes and different compaction algorithms. The attempt here is to group these workloads as examples of what might happen with a certain type of backup device and a certain workload.

**Note**: Remember that these formulas and charts are to give you an idea of what you might achieve from a particular backup device. Your data is as unique as your company and the correct backup device solution must take into account many different factors.

The save and restore rates listed in this document were obtained on a dedicated system. A dedicated system is one where the system is up and fully functioning but no other users or jobs are running except the save and restore operations. All processors and Memory were dedicated to the system and no partial processors were used. Other subsystems such as QBATCH are required in order to run concurrent and parallel operations. All workloads were deleted before restoring them again.

## 15.5 Lower Performing Backup Devices

With the lower performing backup devices, the devices themselves become the gating factor so the save rates are approximately the same, regardless of system CPU size (DVD-RAM).

*Table 15.5.1 Lower performing backup devices LossFromWorkLoadType Approximations (Save Operations)*

| Workload Type | Amount of Loss |
|---|---|
| Large Database File | 95% |
| User Mix / Domino / Network Storage Space | 55% |
| Source File / 1 Directory Many Objects / Many Directories Many Objects | 25% |

**Example for a DVD-RAM:**

**DeviceSpeed * LossFromWorkLoad * Compaction Factor**

0.75 * 0.95 = (.71) * 2.8 = (1.995) MB/S * 3600 = 7182 MB/HR = 7 GB/HR

0.75 * 0.95 = (.71) * No Compression * 3600 = 2556 MB/HR = 2.5 GB/HR

## 15.6 Medium & High Performing Backup Devices

Medium & high performing backup devices (SLR60, SLR100, VXA-2, VXA-320).

*Table 15.6.1 Medium performing backup devices LossFromWorkLoadType Approximations (Save Operations)*

| Workload Type | Amount of Loss |
|---|---|
| Large Database File | 95% |
| User Mix / Domino / Network Storage Space | 65% |
| Source File / 1 Directory Many Objects / Many Directories Many Objects | 25% |

**Example for SLR100:**

**DeviceSpeed * LossFromWorkLoad * Compaction Factor**

5.0 * 0.95 = (4.75) * 2.0 = (9.5) MB/S * 3600 = 34200 MB/HR = 34 GB/HR

## 15.7 Ultra High Performing Backup Devices

High speed backup devices are designed to perform best on large files. The use of multiple high speed backup devices concurrently or in parallel can also help to minimize system save times. See section on Multiple backup devices for more information (3580 Ultrium-2, 3580 Ultrium-3 (2Gb & 4Gb Fiber Channel), 3592J, 3592E).

*Table 15.7.1 Higher performing backup devices LossFromWorkLoadType Approximations (Save Operations)*

| Workload Type | Amount of Loss |
|---|---|
| Large Database File | 95% |
| User Mix / Domino / Network Storage Space | 50% |
| Source File / 1 Directory Many Objects / Many Directories Many Objects | 5% |

**Example for 3580 ULTRIUM-2 Fiber:**

**DeviceSpeed * LossFromWorkLoad * Compaction Factor**

**LG File** 35.0 * 0.95 = (33.25) * 2.8= (93.1) MB/S *3600 = 335160 MB/HR = 335 GB/HR

**UserMix** 35.0 * 0.50 = (17.5) * 2.8= (49) MB/S *3600 = 176400 MB/HR = 176 GB/HR

**Source** 35.0 * 0.05 = (1.75) * 2.8= (4.9) MB/S *3600 = 17640 MB/HR = 17.6 GB/HR

**NOTE: Actual performance is data dependent, these formulas are for estimating purposes and may not match actual performance on customer systems.**

## 15.8 The Use of Multiple Backup Devices

**Concurrent Saves and Restores** - The ability to save or restore different objects from a single library/directory to multiple backup devices or different libraries/directories to multiple backup devices at the **same time** from **different jobs**. The workloads that were used for the testing were Large Database File and User Mix from libraries. For the tests multiple identical libraries were created, a library for each backup device being used.

**Parallel Saves and Restores** - The ability to save or restore a **single object** or library/directory across **multiple backup devices** from the **same job**. Understand that the function was designed to help those customers, with very large database files which are dominating the backup window. The goal is to provide them with options to help reduce that window. Large objects, using multiple backup devices, using the parallel function, can greatly reduce the time needed for the object operation to complete as compared to a serial operation on the same object.

Concurrent operations to multiple backup devices will probably be the preferred solution for most customers. The customers will have to weigh the benefits of using parallel verses concurrent operations for multiple backup devices in their environment. The following are some thoughts on possible solutions to save and restore situations. Remember that memory, processors and DASD play a large factor in whether or not you will be able to make use of parallel or concurrent operations that can be used to affect the back up window.

- For save and restore with a User Mix or small to medium object workloads, the use of concurrent operations will allow multiple objects to be processed at the same time from different jobs, making better use of the backup devices and the system.

- For systems with a large quantity of data and a few very large database files whether in libraries or directories, a mixture of concurrent and parallel might be helpful. (Example: Save all of the libraries/directories to one backup device, omitting the large files from the library or the directory the file is located in. At the same time run a parallel save of those large files to multiple backup devices.)

- For systems dominated by Large Files the only way to make use of multiple backup devices is by using the parallel function.

- For systems with a few very large files that can be balanced over the backup devices, use concurrent saves.

- For backups where libraries/directories increase or decrease in size significantly throwing concurrent saves out of balance constantly, the customer might benefit from the parallel function as the libraries/directories would tend to be balanced against the backup devices no matter how the libraries change. Again this depends upon the size and number of data objects on the system.

- Customers planning for future growth where they would be adding backup devices over time, might benefit by being able to set up Backup Recovery Media Services (BRMS/400) using *AVAIL for backup devices. Then when a new backup device is added to the system and recognized by BRMS/400 it will be used, leaving the BRMS/400 configuration the same but benefiting from the additional backup device. Also the same is true in reverse: If a backup device is lost, the weekly backup doesn't have to be postponed and the BRMS/400 configuration doesn't need to change, the backup will just use the available backup devices at the time of the save.

## 15.9 Parallel and Concurrent Library Measurements

This section discusses parallel and concurrent library measurements for tape drives, while sections later in this chapter discuss measurements for virtual tape drives.

### 15.9.1 Hardware (2757 IOAs, 2844 IOPs, 15K RPM DASD)

**Hardware Environment.**
This testing consisted of an 840 24 way system with 128 GB of memory. The model 840 doesn't support the 15K RPM DASD in the main tower so only 4, 18 GB 10K RPM RAID protected DASD units were in the main tower.

15 PCI-X towers (5094 towers), were attached and filled with 45, 35 GB 15K RPM RAID protected DASD units. 2757 IOAs in all 15 towers and 2844 IOPs. All of the towers attached to the system were configured into 8 High Speed Link (HSL) with two towers in each link. One 5704 fiber channel connector in each tower, or two per HSL. A total of 679 DASD, 675 of which were 35 GB 15K RPM DASD units all in the system ASP. We used the new high speed ULTRIUM GEN 2 tape drives, model 3580 002 fiber channel attached.

There were a lot of different options we could have chosen to try to view this new hardware, we were looking for a reasonable system to get the maximum data flow, knowing that at some point someone will ask what is the maximum. As you look at this information you will need to put it in prospective of your own system or system needs.

We chose 8 HSLs because our bus information would tell us that we can only flow so much data across a single HSL. The total number of 3580 002 tape drives we believe we could put on a link was something a little greater than 2, but the 3rd tape drive would probably be slowed greatly by what the HSL could support, so to maximize the data flow we chose to put only two on a HSL.

What does this mean to your configuration? If you are running large file save and restore operations we would recommend only 2 high speed tape drives per HSL. If your data leans more toward user mix you could probably make use of more drives in a single HSL. How many will depend upon your data. Remember there are other factors that affect save and restore operations, like memory, number of processors available, number and type of DASD available to feed those tape drives, and type of storage IOAs being used.

Large File operations create a great deal of data flow without using a lot of processing power but User Mix data will need those Processors, memory and DASD. Could the large file tests have been done by fewer processors? Yes, probably by something between 8 and 16 but in order to also do the user mix in the same environment we choose to have the 24 processors available. The user mix is a more generic customer environment and will be informational to a larger set of customers and we wanted to be able to provide some comparison information for most customers to be able to use here.

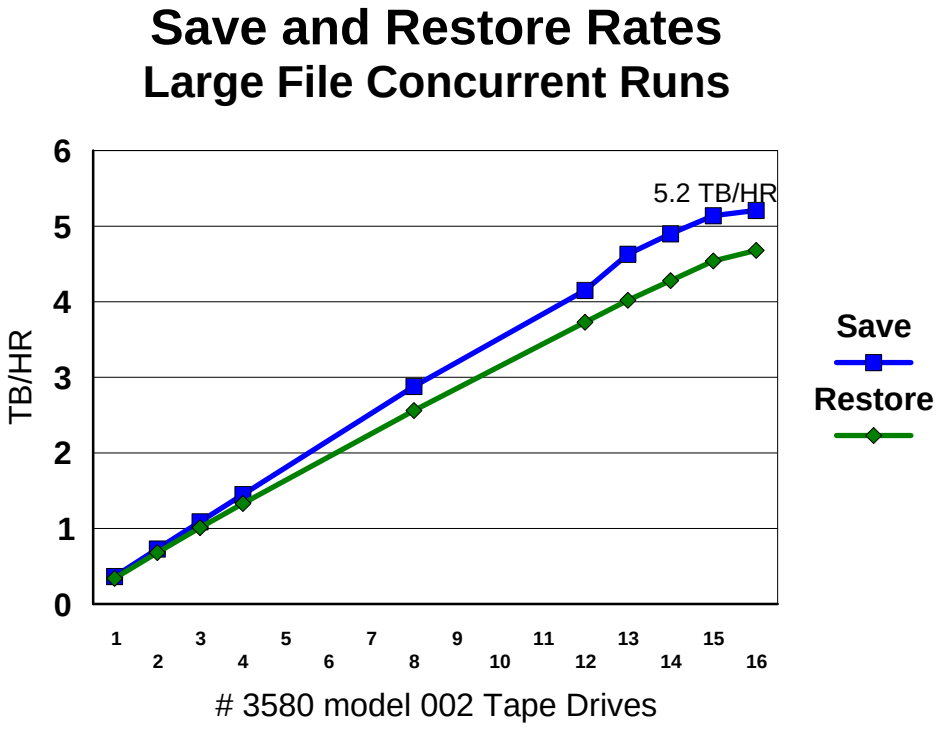
### 15.9.2 Large File Concurrent

For the concurrent testing 16 libraries were built, each containing a single 320 GB file with 80 4 GB members. The file size was chosen to sustain a flow across the HSL, system bus, processors, memory and tapes drives for about an hour. We were not interested in peak performance here but sustained performance. Measurements were done to show scaling from 1 to 16 tape drives, knowing that near the top number of tape drives that the system would become the limiting factor and not the tape drives. This could be used by customers to give them an estimate at what might be a reasonable number of tape drives for their situation.

*Table 15.9.2.1 iV5R2 16 - 3580.002 Fiber Channel Tape Device Measurements (Concurrent) (Save = S, & Restore = R)*

| # 3580.002 Tape drives | | 1 | 2 | 3 | 4 | 8 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 320 GB DB file with 80 4 GB members | S | 365 GB/HR | 730 GB/HR | 1.09 TB/HR | 1.45 TB/HR | 2.88 TB/HR | 4.15 TB/HR | 4.63 TB/HR | 4.90 TB/HR | 5.14 TB/HR | 5.21 TB/HR |
| | R | 340 GB/HR | 680 GB/HR | 1.01 TB/HR | 1.33 TB/HR | 2.56 TB/HR | 3.73 TB/HR | 4.02 TB/HR | 4.28 TB/HR | 4.54 TB/HR | 4.68 TB/HR |

In the table above, you will notice that the 16th drive starts to loose value. Even though there is gain we feel we are starting to see the system saturation points start to factor in. Unfortunately, we didn't have anymore drives to add in but believe that the total data throughput would be relatively equal, even if any more drives were added.

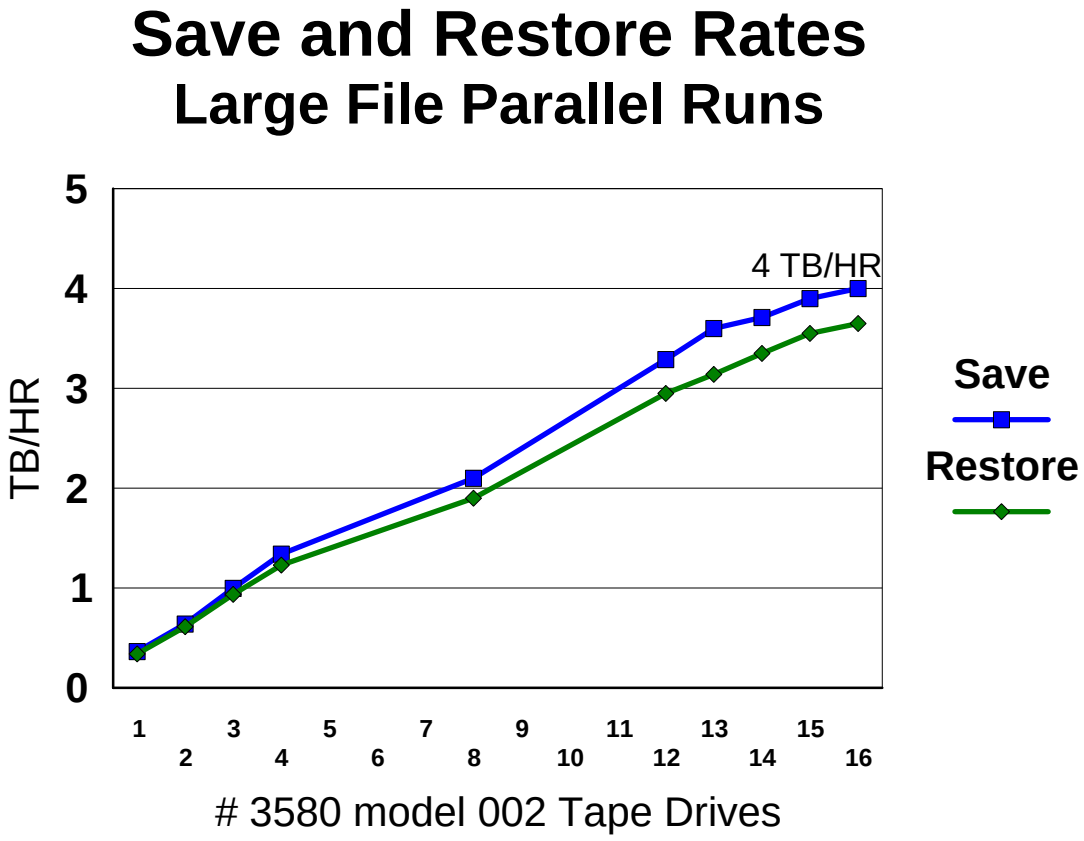


### 15.9.3 Large File Parallel

For the measurements in this environment, BRMS was used to manage the save and restore, taking advantage of the ability built into BRMS to split an object between multiple tape drives. Starting with a 320 GB file in a single library and building it up to 2.1 TB for tape drive tests 1 - 4 and 8. The file was then duplicated in the library for tape drive tests 12 - 16, a single library with two 2.1 TB files was used. Not quite the same as having a 4.2 TB file. Because of certain limitations in building our test data, we felt this was the best way to build the test data. The goal is to see scaling of tape drives on the system along with trying to locate any saturation points that might help our customers identify limitations in their own environment.

*Table 15.9.3.1 iV5R2 16 - 3580.002 Fiber Channel Tape Device Measurements (Parallel)*
*(Save = S, & Restore = R)*

| # 3580.002 Tape drives | 1 | 2 | 3 | 4 | | 8 | | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S | 363 GB/HR | 641 GB/HR | 997 GB/HR | 1.34 TB/HR | | 2.1 TB/HR | | 3.29 TB/HR | 3.60 TB/HR | 3.71 TB/HR | 3.90 TB/HR | 4 TB/HR |
| R | 340 GB/HR | 613 GB/HR | 936 GB/HR | 1.23 TB/HR | | 1.90 TB/HR | | 2.95 TB/HR | 3.14 TB/HR | 3.35 TB/HR | 3.55 TB/HR | 3.65 TB/HR |

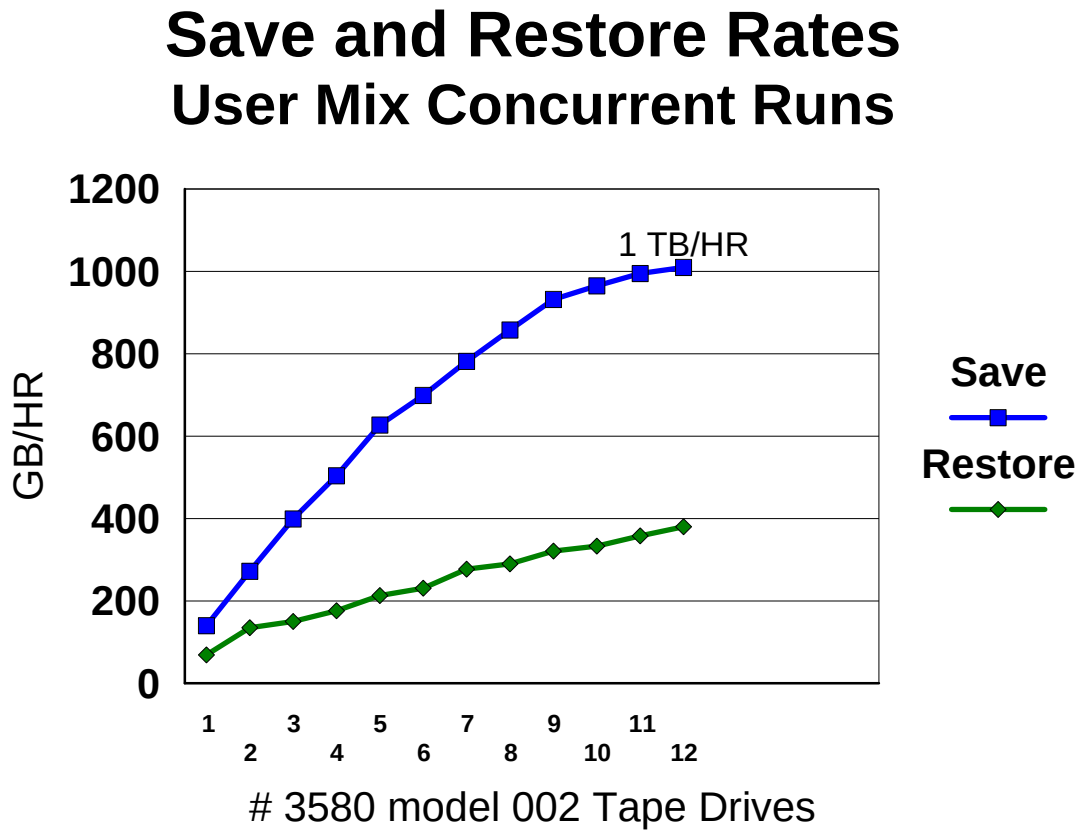


### 15.9.4 User Mix Concurrent

User Mix will generally portray a fair population of customer systems, where the real data is a mixture of programs, menus, commands along with their database files. The new ultra tape drives are in their glory when streaming large file data, but a lot of other factors play a part when saving and restoring multiple smaller objects.

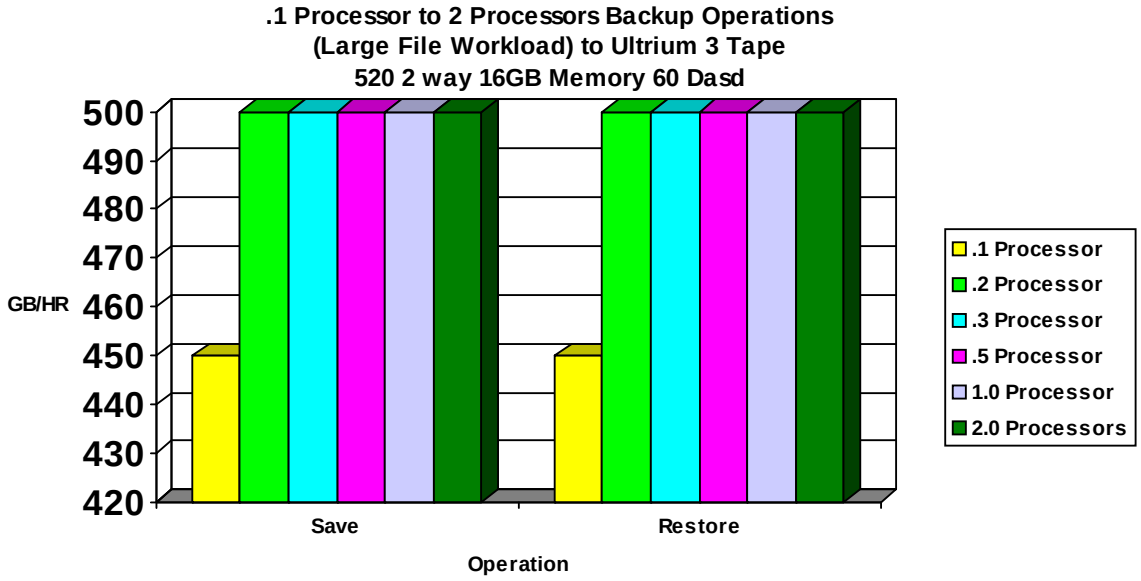
*Table 15.9.4.1 iV5R2 16 - 3580.002 Fiber Channel Tape Device Measurements (Concurrent) (Save = S, & Restore = R)*

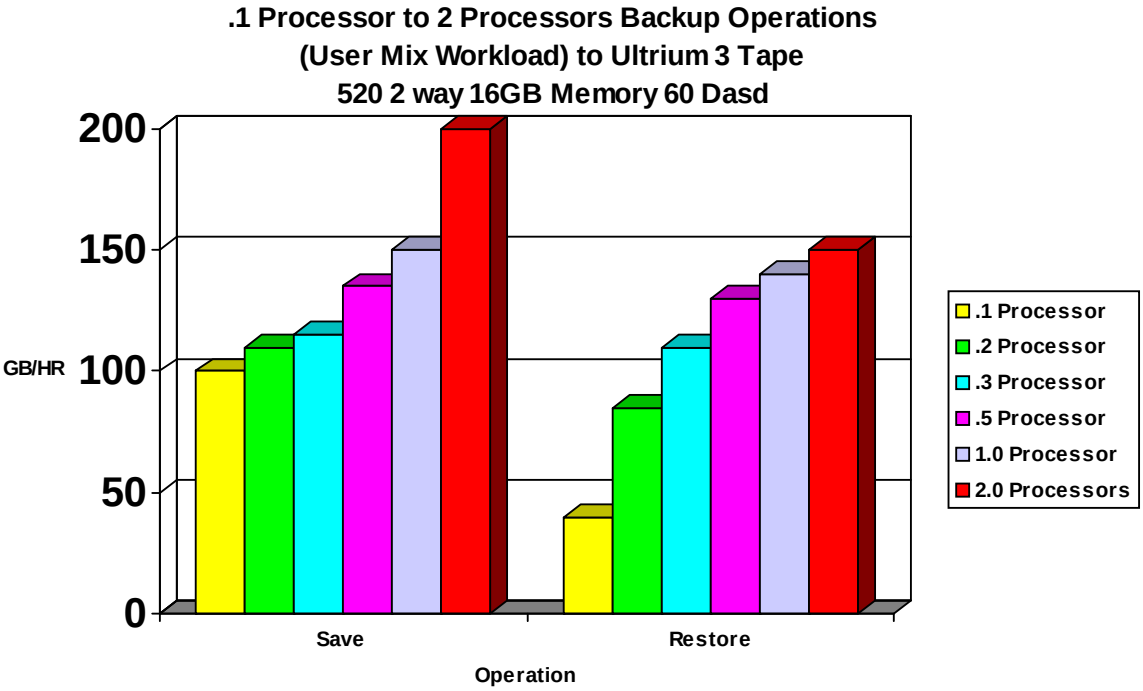
| # 3580.002 Tape drives | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 12 GB total Library size workload was used for modeling this, as described in section 15.3 | S | 140 GB/HR | 272 GB/HR | 399 GB/HR | 504 GB/HR | 627 GB/HR | 699 GB/HR | 782 GB/HR | 858 GB/HR | 932 GB/HR | 965 GB/HR | 995 GB/HR | 1010 GB/HR |
| | R | 69 GB/HR | 135 GB/HR | 150 GB/HR | 176 GB/HR | 213 GB/HR | 231 GB/HR | 277 GB/HR | 290 GB/HR | 321 GB/HR | 333 GB/HR | 358 GB/HR | 380 GB/HR |



## 15.10 Number of Processors Affect Performance

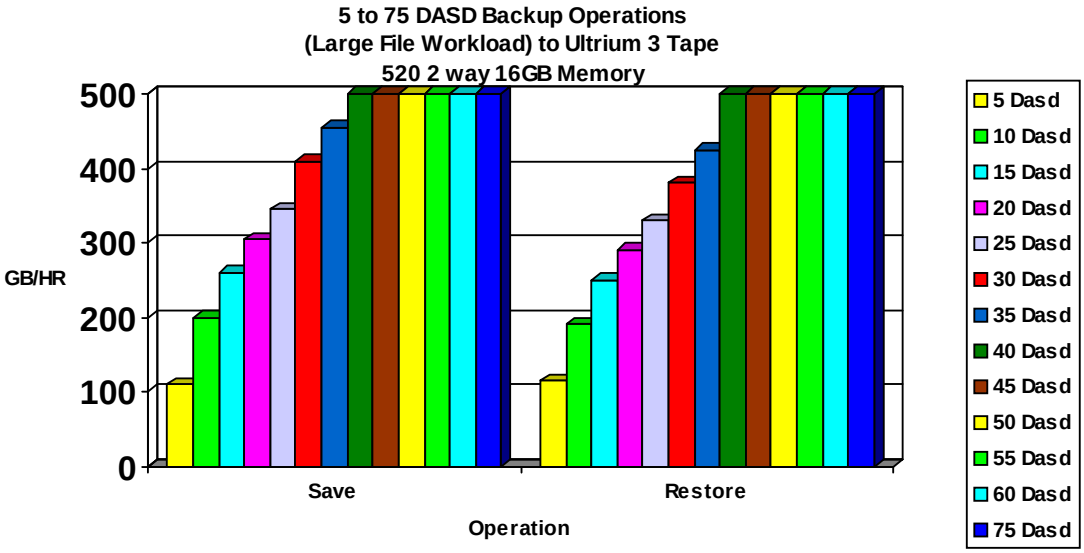
With the Large Database File workload, it is possible to fully feed two backup devices with a single processor, but with the User Mix workload it takes 1+ processors to fully feed a backup device. A recommendation might be 1 and 1/3 processors for each backup device you want to feed with User Mix data.

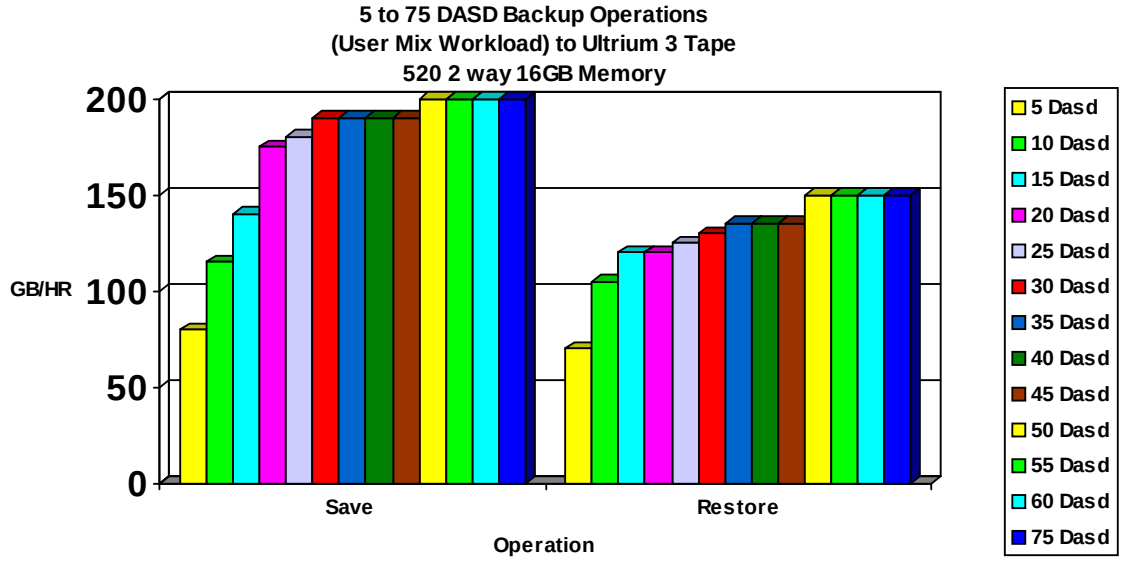


## 15.11 DASD and Backup Devices Sharing a Tower

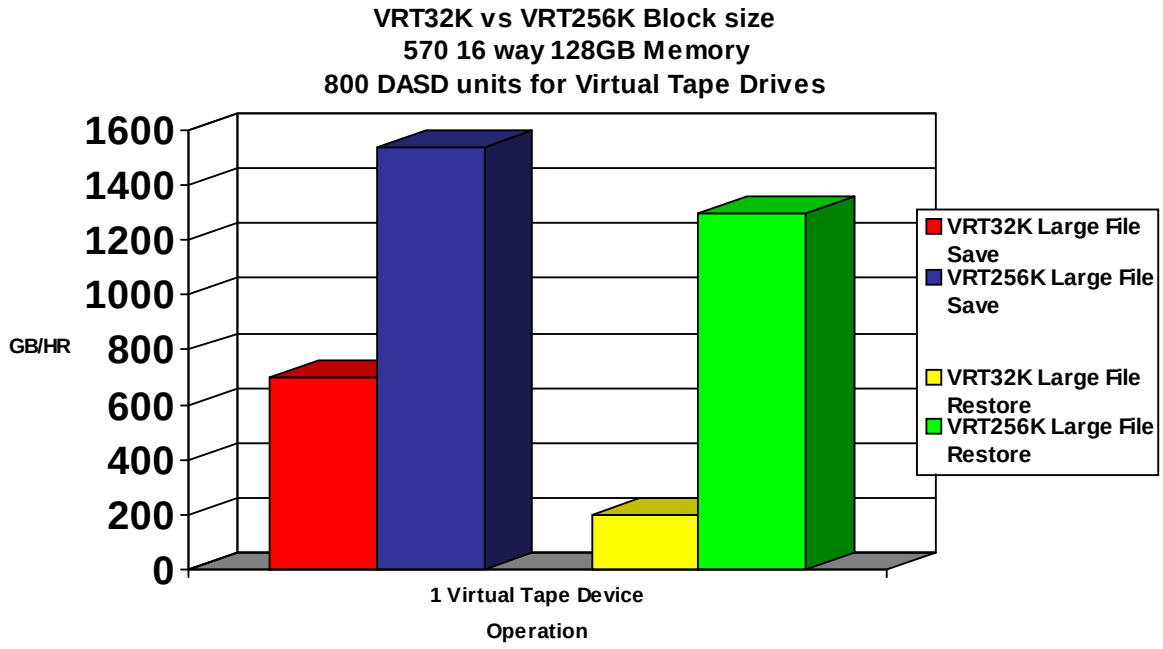
The system architecture does not require that DASD and backup devices be kept separated. Testing in the IBM Rochester Lab, we had attached one backup device to each tower and all towers had 45 DASD units in them, when we did the 3580 002 testing. The 3592J has similar characteristics to the 3580 002 but the 3580 003 and 3592E models have greater capacities which create new scenarios. You aren't physically limited to putting one backup device in a tower, but for the newest high speed backup devices you can saturate the bus if you have multiple devices in a tower. You need to look at your total system or partition configuration in order to determine if it is possible to use multiple high speed devices on the system and still get the most out of these devices. No matter what you determine is possible we advocate spreading your backup devices amongst the towers available.



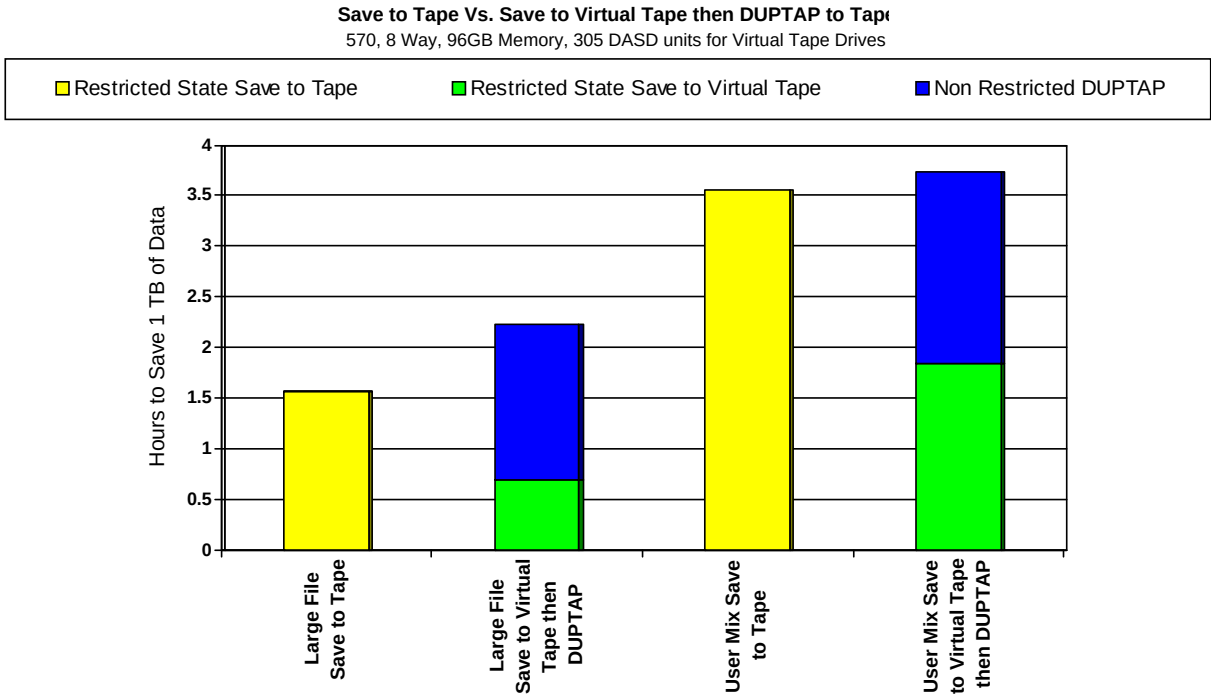



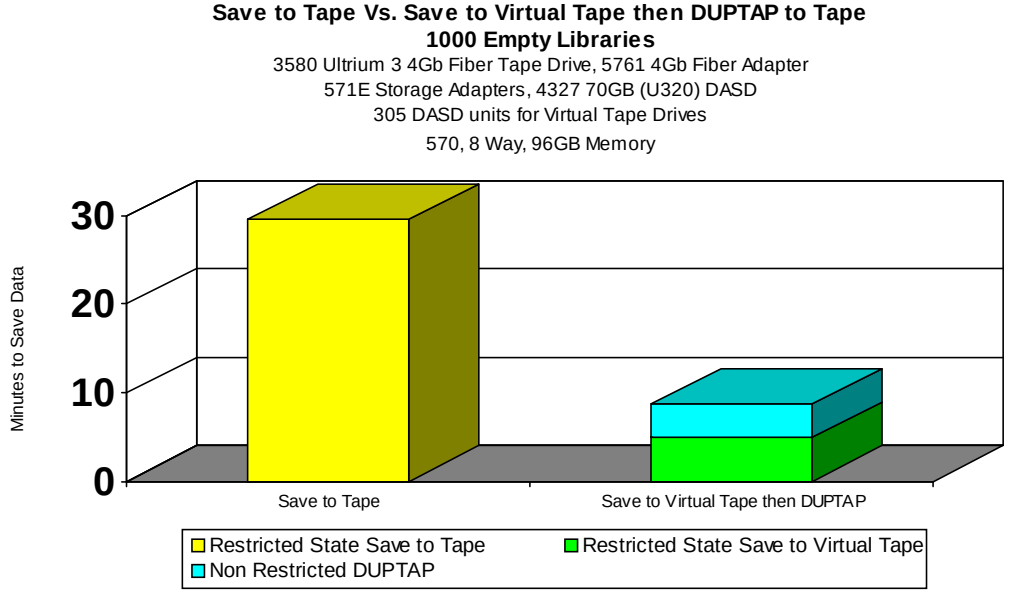
## 15.12 Virtual Tape

Virtual tape drives are being introduced in iV5R4 so those customers can make use of the speed of saving to DASD, then save the data using DUPTAP to the tape drives reducing the backup window where the system is unavailable to users. There are a lot of pieces to consider in setting up and using Virtual tape drives. The block size must match the physical backup device block capabilities you will be using. The following helps to show that even if your workload is large file you may not gain anything in your back up window even using the virtual tape drives. If your tape drive uses smaller block sizes your virtual tape drive must use small blocks



The following measurements were done on a system with newer hardware including a 3580 Ultrium 3 4Gb Fiber Channel Tape Drive, 571E storage adapters, and 4327 70GB (U320) DASD.

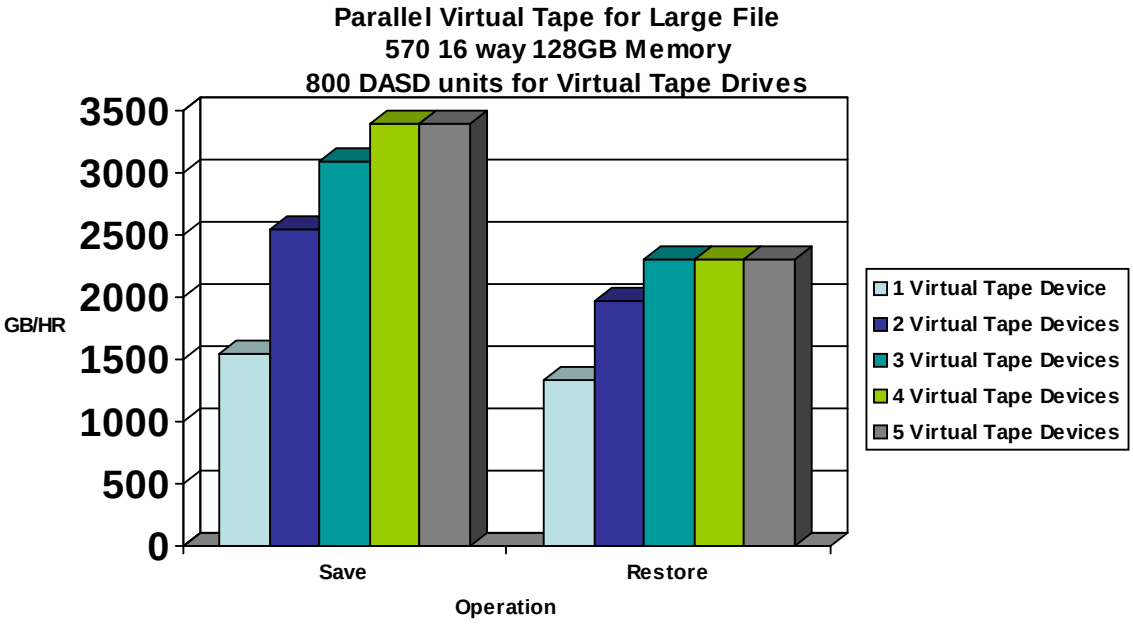


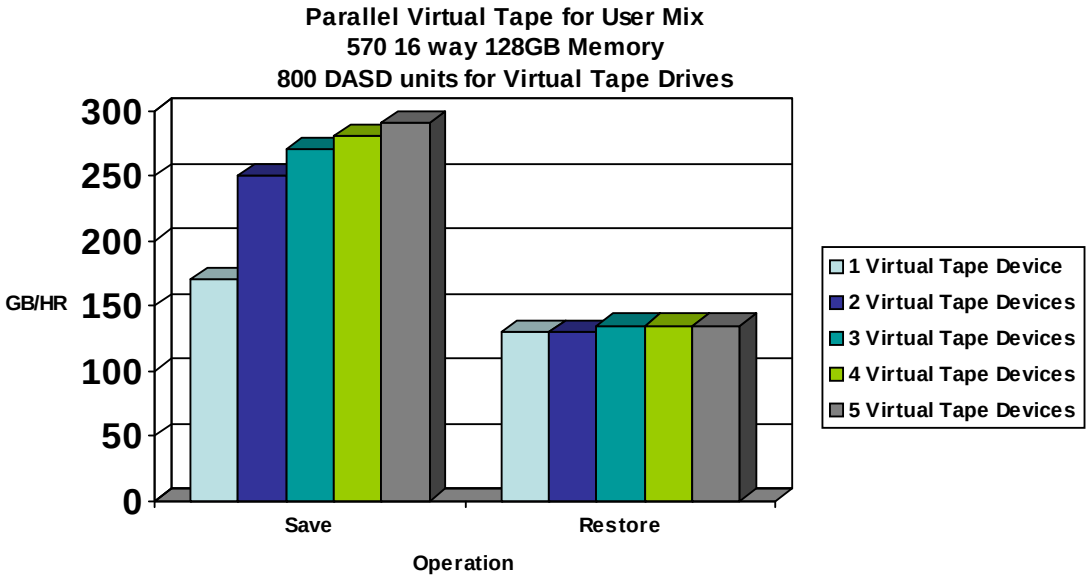




Measurements were also done comparing save of 1000 empty libraries to tape versus save of these libraries to virtual tape followed by DUPTAP from the virtual tape to tape. The save to tape was much slower which can be explained as follows. When data is being saved to tape, a flush buffer is requested after each file is written to ensure that the file is actually on the tape. This forces the drive to backhitch for each file and greatly reduces the performance. The DUPTAP command does not need to send a flush buffer until the duplicate command completes, so it does not have the same performance impact.

## 15.13 Parallel Virtual Tapes

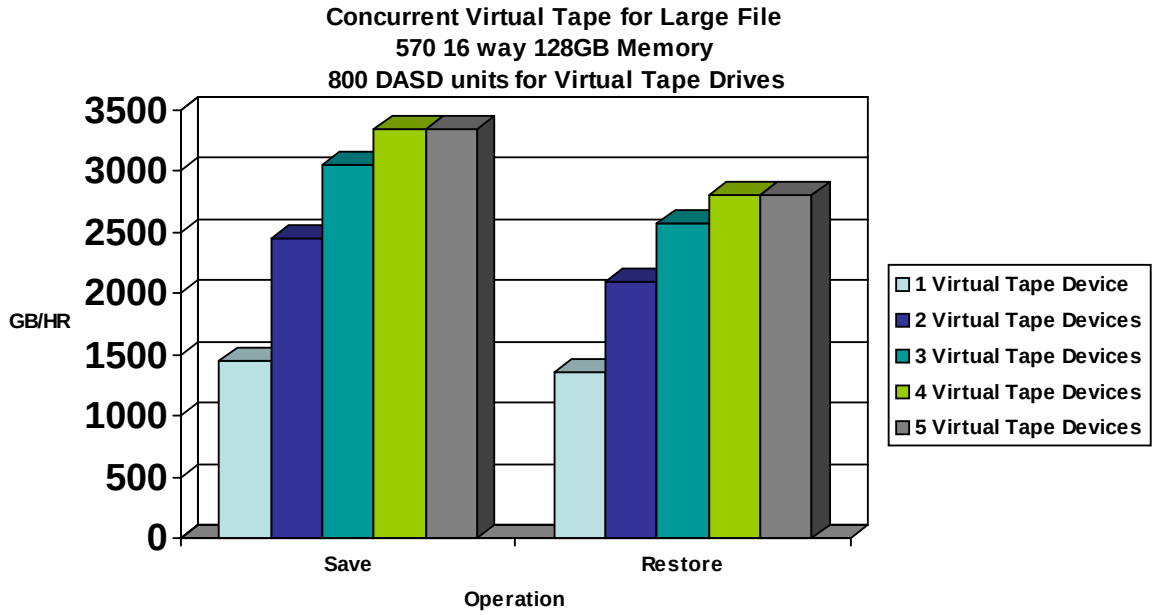
NOTE: Virtual tape is reading and writing to the same DASD so the maximum throughput with our concurrent and parallel measurements is different than our tape drive tests where we were reading from DASD and writing to tape.

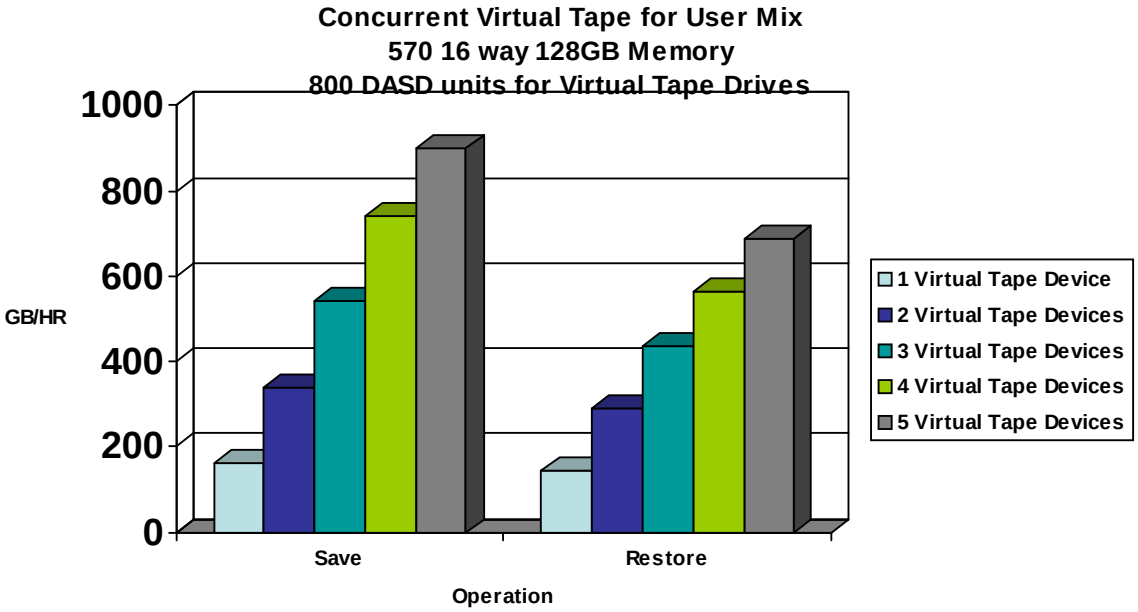


## 15.14 Concurrent Virtual Tapes

NOTE: Virtual tape is reading and writing to the same DASD so the maximum throughput with our concurrent and parallel measurements is different than our tape drive tests where we were reading from DASD and writing to tape.

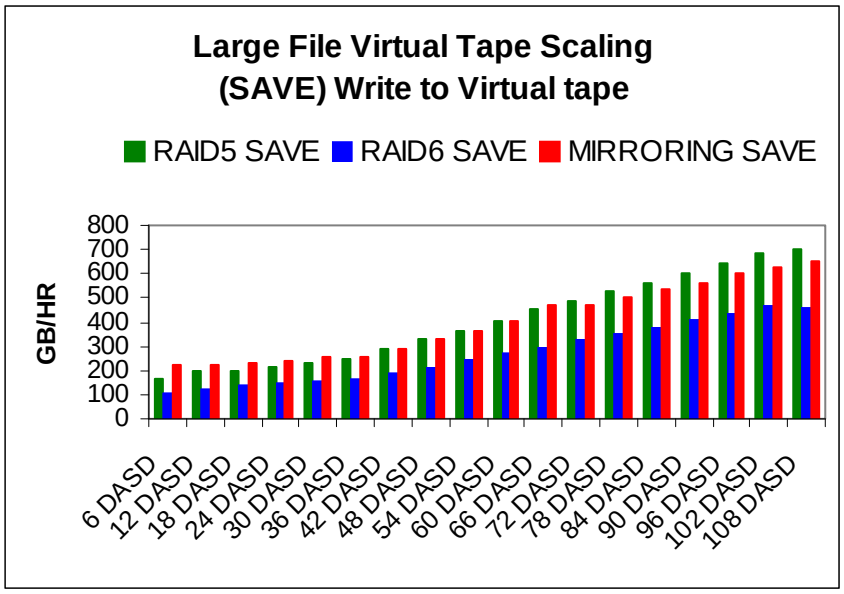


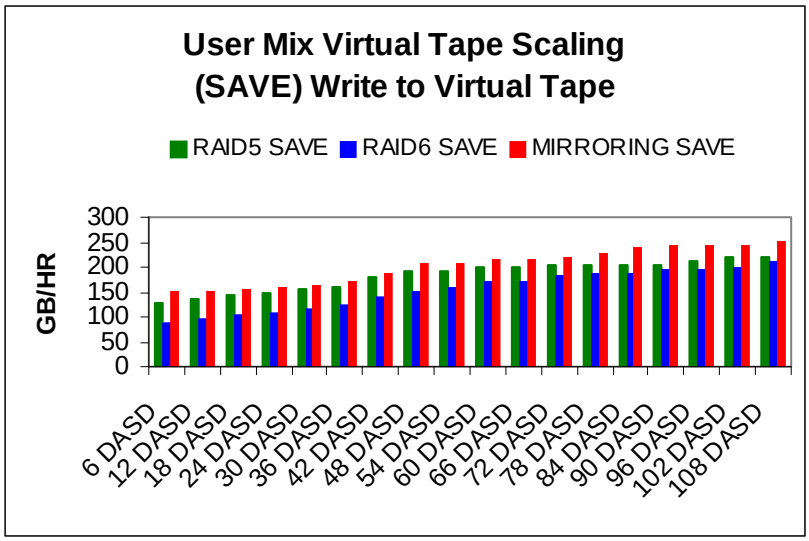



## 15.15 Save and Restore Scaling using a Virtual Tape Drive.

A 570 8 way System i was used for the following tests. A user ASP was created using up to 3 571F IOAs with up to 36 U320 70 GB DASD on each IOA. The Chart shows the number of DASD in each test and the Virtual tape drive was created using that DASD.

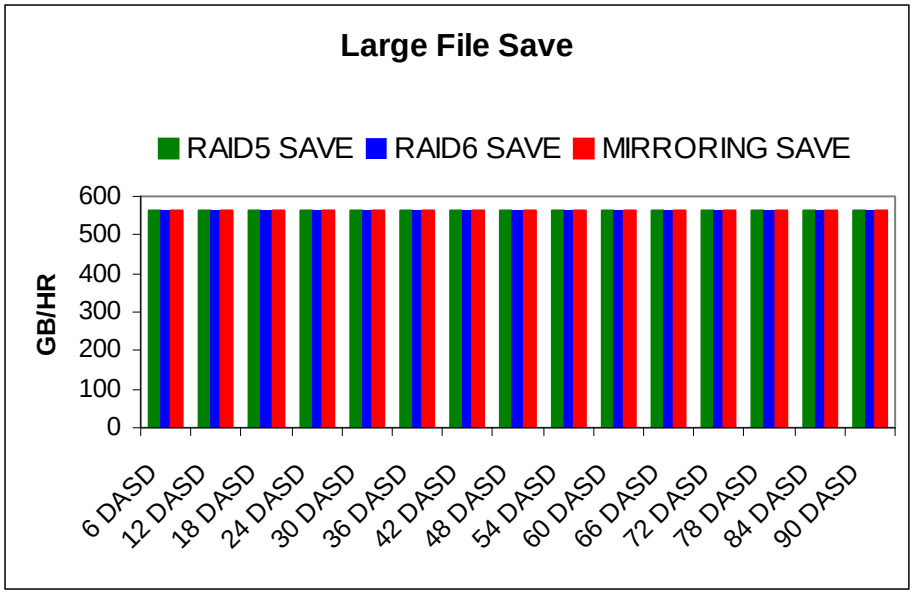
The workload data was restored into the system ASP and was then saved to the Virtual tape drive in the user ASP. The system ASP consisted of 2 HSL loops, a mix of 571E and 571F IOAs and 312 - 70GB U320 DASD units. These charts are very specific to this DASD but the scaling flow would be similar with different IOAs the actual rates would vary. For more information on the IOAs and DASD see Chapter 14 of this guide. Restoring the workloads from the Virtual tape drives started at 900 GB/HR reading from 6 DASD and scaled up to 1.5 TB/HR on the 108 DASD. The bottle neck will be limited to where you are writing and how many DASD are available to the write operation.

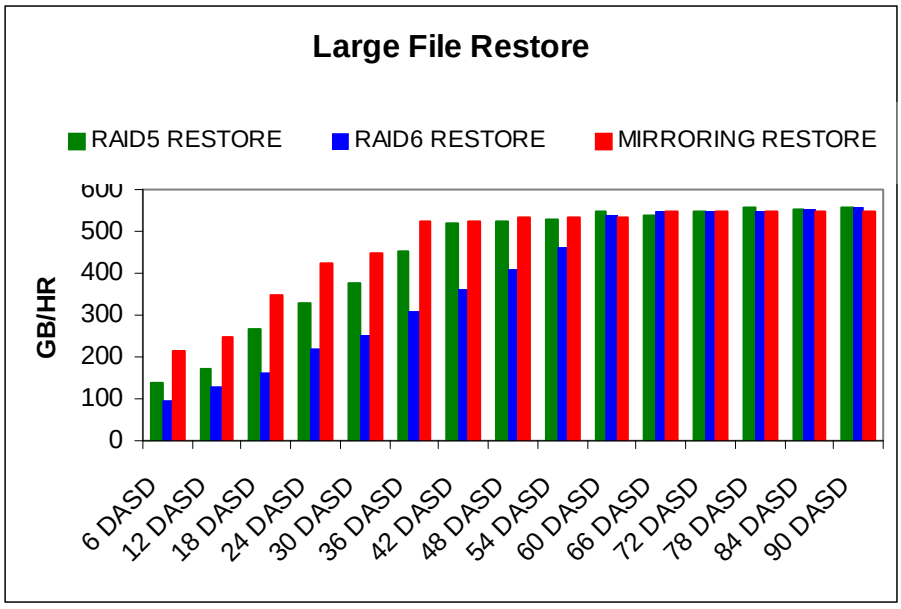


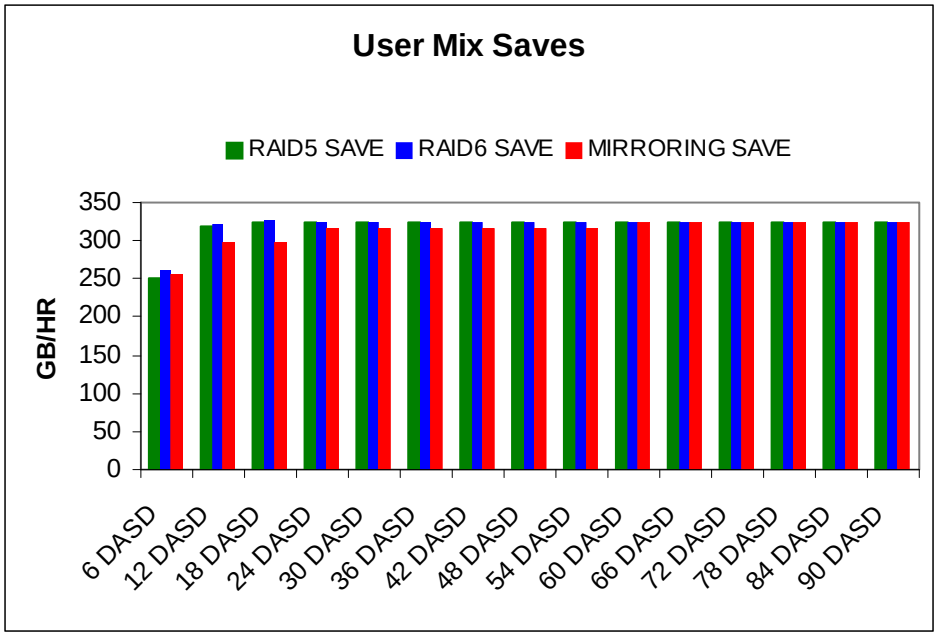



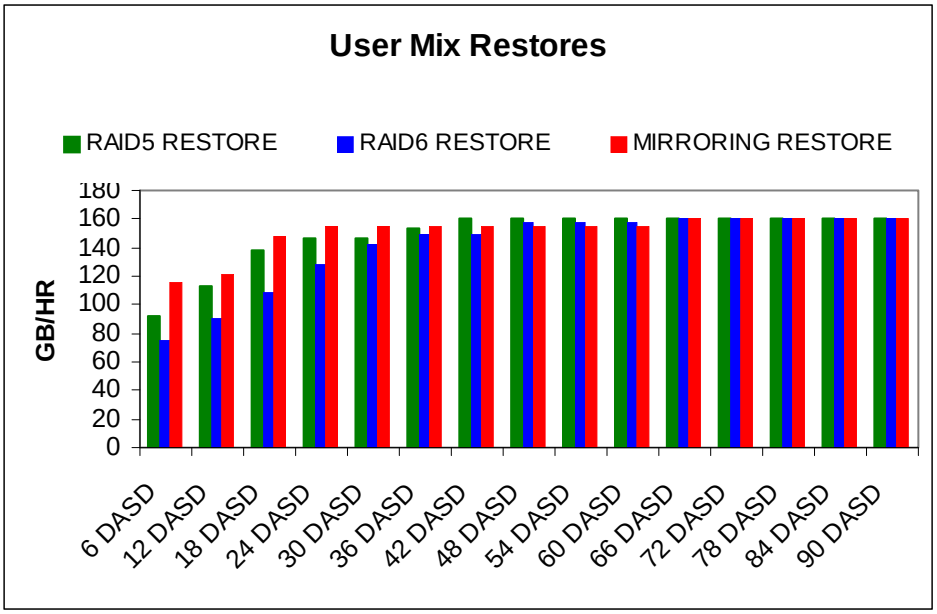
## 15.16 Save and Restore Scaling using 571E IOAs and U320 15K DASD units to a 3580 Ultrium 3 Tape Drive.

A 570 8 way System i was used for the following tests. A user ASP was created with the number of DASD listed in each test . The workload data was then saved to the tape drive , deleted from the system and restored to the user ASP. These charts are very specific to the new IOAs and U320 capable DASD available. For more information on the IOAs and DASD see Chapter 14 of this guide.





## 15.17 High-End Tape Placement on System i

The current high-end tape drives (ULTRIUM-2 / ULTRIUM-3 and 3592-J / 3592-E) need to be placed carefully on the System i buses and HSLs in order to avoid bottlenecking. The following rules-of thumb will help optimize performance in a large-file save environment, and help position the customer for future growth in tape activity:

- Limit the number of drives per fibre tape adapter as follows:
  - For ULTRIUM-2, 3592-J, and slower drives, two drives can share a fc 5704 or fc 5761 fibre tape adapter. If running on a 2 GByte loop, a 3rd drive can share a fc 5761 fibre tape adapter
  - For ULTRIUM-3 and TS1120 (3592-E) drives, each drive should be on a separate fibre tape adapter

- Place the fc 5704 or fc 5761 in a 64-bit slot on a "fast bus" as follows:
  - PCI-X
    - In a 5094/5294 tower use slot C08 or C09.
    - In a 5088/0588 tower use slot C08 or C09. You may need to purchase RPQ #847204 to allow the tower to connect with RIO-G performance
    - In an 0595 or 5095 or 5790 expansion unit, use any valid slot
  - PCI
    - In a 5074/5079/5078 tower, use slot C02, C03 or C04
  - Note
    - "Ensure the fc 5761 is supported on your CPU type"

- Put one fc 5704 or fc 5761 per tower initially. On loops running at 2 GByte speeds, a second fc 5704 card can be added according to the locations recommended above if needed.

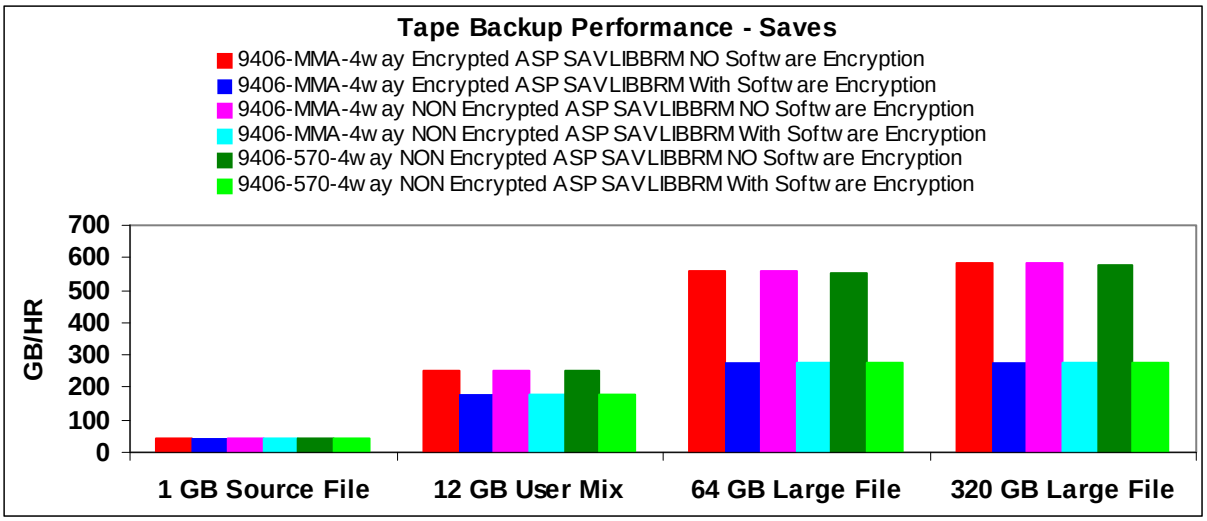
- Spread tape fibre cards across as many HSL's as possible, with maximums as follow
  - On Loops running at 1 GByte (e.g. all loops on 8xx systems, or loops with HSL-1 towers )
    - Maximum of two drives per HSL loop
  - On Loops running at 2 GByte (eg loops with all HSL-2 / RIO-G towers on system i systems)
    - Maximum of six ULTRIUM-2 or 3592-J drives per RIO-G loop.
    - Maximum of four ULTRIUM-3 drives or TS1120 (3592-E) drives per RIO-G loop using the fc 5704 IOA.
    - Maximum of two TS1120 (3592-E) drives per RIO-G loop using the fc 5761 IOA

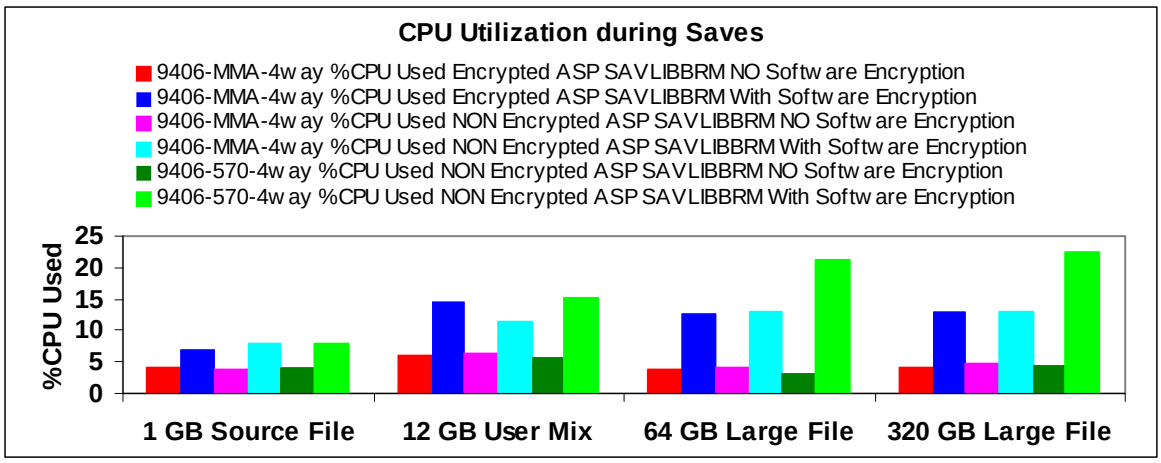
- If Gbit Ethernet cards are present on the system and will be running during the backups, then treat them as though they were ULTRIUM-3 or TS1120 (3592-E) tape drives when designing the card and HSL placement using the rules above since they can command similar bandwidth

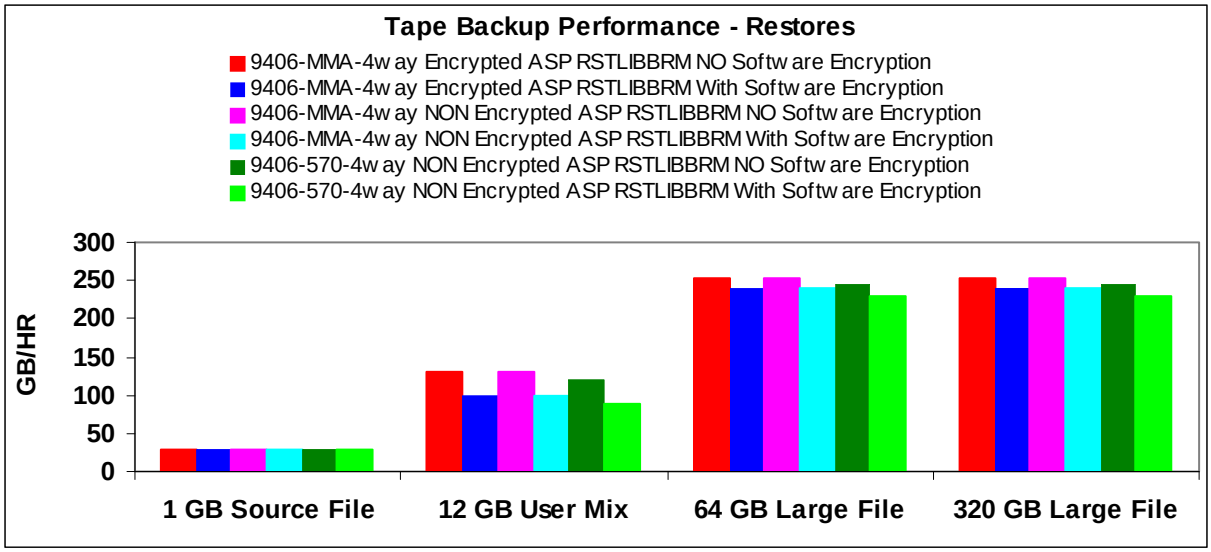
The rules above assume that the customer is running a large-file workload and that all tape drives are active simultaneously. If your customer is running a user-mix tape workload or the high load cards are not running simultaneously, then it may be possible to put more gear on the bus/HSL than shown. There may also be certain card layouts that will allow more drives per bus/tower/HSL, but these need to be reviewed individually.

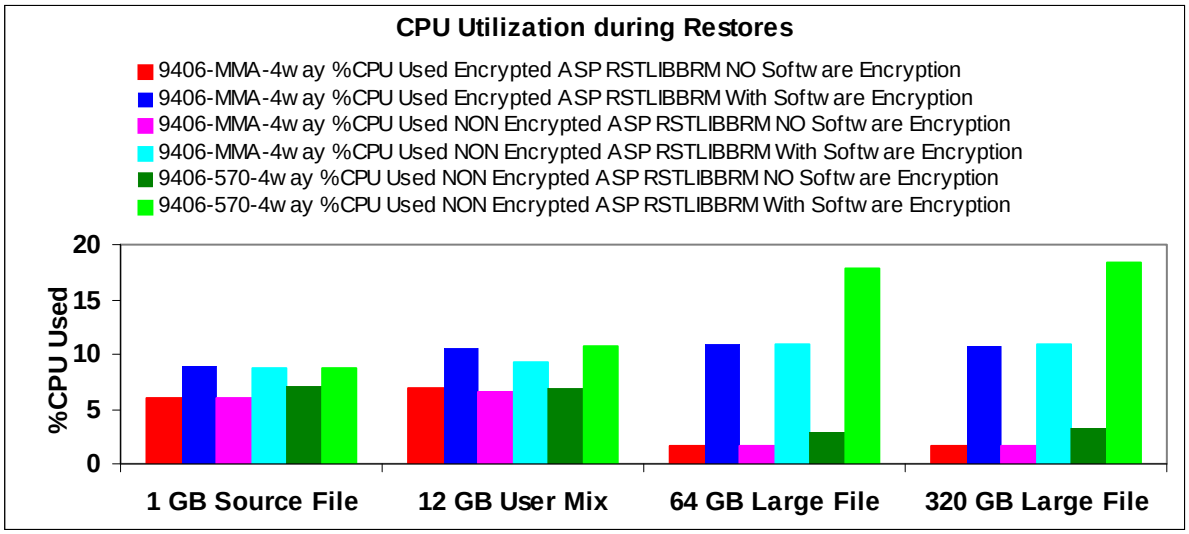
## 15.18 BRMS-Based Save/Restore Software Encryption and DASD-Based ASP Encryption

The Ultrium-3 was used in the following experiments, which attempt to characterize the effects of BRMS-based save /restore software encryption and DASD-based ASP encryption. Some of the newer tape drives offer hardware encryption as an option but for those who are not looking to upgrade or invest in these tape units at this time, software encryption can be a fair solution. In the experiments we used full processors that were dedicated to the partition. We used a 9406-MMA 4 way partition and a 9406-570 4 way partition. Both systems had 40 GB of main memory. The workload data was located on a single user ASP with 36 - 70GB 15K RPM DASD attached through a 571F IOA. The experiments were not set up to show the best possible environment or to take into account all of the possible hardware environments, instead these experiments were an attempt to portray some of the differences customers might observe if they choose software encryption as a back up strategy over their current non-encrypted environment. Software encryption has a significant impact on save times but only a minor impact to restore times






Performance will be limited to the native drive rates (shown in table 15.1.1) because encrypted data blocks have a very low compaction ratio.

## 15.19 5XX Tape Device Rates

Note: Measurements for the high speed devices were completed on a 570 4 way system with 2844 IOPs and 2780 IOA's and 180 15K RPM RAID5 DASD units. The smaller tape device tests were completed on a 520 2 way with 75 DASD units. The Virtual tape and *SAVF runs were completed on a 570 ML16 with 256GB of memory and 924 DASD units. The goal of each of the tests is to show the capabilities of the device and so a system large enough to achieve the maximum throughput for that device was used. Customer performance will be dependent on over all systems resources and if those resources match the maximum capabilities of the device. See other sections in this guide about memory, CPU and DASD.

*Table 15.19.1*
*Measurements in (GB/HR) Workload data Saved and Restored from User ASP 2.*

| Workload<br>S = Save<br>R = Restore | | SLR60 | SLR100 | VXA-2 | 6279 VXA-320 | | 5755<br>½ High ULTRIUM 2 | 3580 Ultrium 2<br>5704 2GB Fiber Adapter | 3592J<br>5704 2GB Fiber Adapter | 3580 Ultrium 3<br>5704 2GB Fiber Adapter | 3592E Fiber<br>5704 2GB Fiber Adapter | 3592E Fiber<br>fc 5761 4GB Fiber Adapter | *SAVF | Virtual Tape Drive |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Release Measurements were done | | iV5R4 | iV5R4 | iV5R3 | iV5R4 | | iV5R4 | iV5R3 | iV5R3 | iV5R4 | iV5R4 | iV5R4 | iV5R3 | iV5R4 |
| Source File 1GB | S | 17 | 17 | 14 | 19 | | 21 | 17 | 17 | 22 | 30 | 30 | 35 | 35 |
| | R | 19 | 17 | 19 | 19 | | 24 | 20 | 20 | 29 | 30 | 30 | 20 | 20 |
| User Mix 3GB | S | 30 | 31 | 33 | 48 | | | 113 | 130 | | | | | |
| | R | 30 | 31 | 33 | 48 | | | 50 | 115 | | | | | |
| User Mix 12GB | S | 32 | 35 | 40 | 70 | | 145 | 150 | 180 | 200 | 210 | 210 | 220 | 220 |
| | R | 30 | 31 | 35 | 53 | | 96 | 80 | 120 | 150 | 180 | 180 | 125 | 180 |
| Large File 4GB | S | 32 | 34 | 37 | | | | 280 | 280 | | | | | |
| | R | 32 | 34 | 37 | | | | 280 | 340 | | | | | |
| Large File 32GB | S | | | 41 | 82 | | 225 | 350 | 365 | 500 | 560 | 800 | | |
| | R | | | 40 | 68 | | 175 | 330 | 390 | 500 | 560 | 800 | | |
| Large File 64GB | S | | | | 82 | | 225 | 350 | 365 | 500 | 560 | 830 | 1330 | 1450 |
| | R | | | | 68 | | 175 | 330 | 390 | 500 | 560 | 830 | 1340 | 1500 |
| Large File 320GB | S | | | | | | | | | 525 | 580 | 890 | 1420 | 1700 |
| | R | | | | | | | | | 510 | 570 | 830 | 1340 | 1530 |
| 1 Directory<br>Many Objects | S | 23 | 25 | 27 | 40 | | 35 | 65 | 65 | 65 | 65 | 65 | 65 | 70 |
| | R | 12 | 13 | 13 | 30 | | 47 | 14 | 16 | 50 | 60 | 60 | 50 | 60 |
| Many Directories<br>Many Objects | S | 25 | 25 | 30 | 40 | | 35 | 50 | 50 | 50 | 50 | 50 | 50 | 55 |
| | R | 9 | 9 | 9 | 20 | | 23 | 9 | 9 | 30 | 30 | 30 | 23 | 30 |
| Domino Mail Files | S | 29 | 29 | 35 | 67 | | 125 | 190 | 230 | 410 | 500 | 530 | 1000 | 1250 |
| | R | 29 | 29 | 33 | 55 | | 110 | 190 | 230 | 420 | 500 | 560 | 1000 | 1200 |
| Network Storage Space | S | 34 | 34 | 40 | 70 | | 125 | 200 | 230 | 350 | 380 | 500 | 1100 | 1100 |
| | R | 34 | 34 | 40 | 56 | | 140 | 200 | 260 | 380 | 380 | 490 | 1050 | 1100 |

| *Table 15.19.2 - iV5R4M0 Measurements on an 5XX 1-way system 8 RAID5 protected DASD Units 8 GB memory Measurements in (GB/HR) all 8 DASD in the system ASP .* | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Workload<br>S = Save<br>R = Restore | | 6258 4MM tape Drive | SLR60 from table 15.18.1 | | | | | | | |
| Release Measurements were done | | iV5R4M0 | iV5R4 | | | | | | | |
| Source File 1GB | S | 22 | 17 | | | | | | | |
| | R | 15 | 19 | | | | | | | |
| User Mix 12GB | S | 34 | 30 | | | | | | | |
| | R | 30 | 30 | | | | | | | |
| Large File 32GB | S | 39 | 32 | | | | | | | |
| | R | 37 | 32 | | | | | | | |
| 1 Directory<br>Many Objects | S | 12 | 23 | | | | | | | |
| | R | 8 | 12 | | | | | | | |
| Many Directories<br>Many Objects | S | 15 | 25 | | | | | | | |
| | R | 7 | 9 | | | | | | | |
| Domino Mail Files | S | 15 | 29 | | | | | | | |
| | R | 15 | 29 | | | | | | | |
| Network Storage Space | S | 19 | 34 | | | | | | | |
| | R | 19 | 34 | | | | | | | |

## 15.20 5XX Tape Device Rates with 571E & 571F Storage IOAs and 4327 (U320) Disk Units

Save/restore rates of 3580 Ultrium 3 (2Gb and 4Gb Fiber Channel) tape devices and of virtual tape devices were measured on a 570 8-way system with 571E and 571F storage adapters and 714 type 4327 70GB (U320) disk units. Customer performance will be dependent on overall system resources and how well those resources match the maximum capabilities of the device. See other sections in this guide about memory, CPU and DASD.

*Table 15.20.1*
*Measurements in (GB/HR)*
*Workload data Saved and Restored from User ASP 2.*

| Workload<br>S = Save<br>R = Restore | | 2780 Storage IOAs<br>4326 35GB (U160) DASD<br>(Data from table 15.18.1) | 571E/571F Storage IOAs<br>4327 70GB (U320) DASD | | | |
|---|---|---|---|---|---|---|
| | | 3580 Ultrium 3<br>5704 2Gb Fiber Adapter | 3580 Ultrium 3<br>5704 2Gb Fiber Adapter | 3580 Ultrium 3<br>5761 4Gb Fiber Adapter | 3580 Ultrium 4<br>5761 4Gb Fiber Adapter | Virtual Tape Drive |
| Release Measurements were done | | iV5R4 | iV5R4 | iV5R4 | iV5R4 | iV5R4 |
| Source File 1GB | S | 22 | 95 | 110 | 55 | 110 |
| | R | 29 | 40 | 40 | 26 | 40 |
| User Mix 12GB | S | 200 | 290 | 290 | 295 | 345 |
| | R | 150 | 175 | 175 | 182 | 195 |
| Large File 64GB | S | 500 | 510 | 585 | 650 | 1380 |
| | R | 500 | 550 | 785 | 760 | 1230 |
| Large File 320GB | S | 525 | 525 | 635 | 650 | 1420 |
| | R | 510 | 550 | 785 | 760 | 1240 |
| 1 Directory<br>Many Objects | S | 65 | 80 | 80 | 90 | 80 |
| | R | 50 | 60 | 60 | 45 | 65 |
| Many Directories<br>Many Objects | S | 50 | 55 | 60 | 65 | 65 |
| | R | 30 | 30 | 30 | 25 | 30 |
| Domino Mail Files | S | 410 | 440 | 450 | 550 | 1410 |
| | R | 420 | 460 | 460 | 600 | 1190 |
| Network Storage Space | S | 350 | 355 | 410 | 425 | 1300 |
| | R | 380 | 405 | 460 | 525 | 1230 |

## 15.21 5XX DVD RAM and Optical Library

*Table 15.21.1 - iV5R3 Measurements on an 520 2-way system 53 RAID protected DASD Units 16 GB memory Measurements in (GB/HR) ASP 1 (System ASP 23 DASD) ASP 2 (30 DASD) Workload data Saved and Restored from User ASP 2.*

| Workload S = Save R = Restore | | 6331 DTACPR *NO | 6331 DTACPR *YES | 6333 DTACPR *NO | 6333 DTACPR *YES | 6330 DTACPR *NO | 6330 DTACPR *YES | | 399F Model 200 Optical Library UDO | 399F Model 200 Optical Library 14x |
|---|---|---|---|---|---|---|---|---|---|---|
| Release Measurements were done | | V5R3 | V5R3 | V5R3 | V5R3 | V5R3 | V5R3 | | V5R3 | V5R3 |
| Source File 1GB | S | 1.8 | 9.0 | 2.2 | 12.0 | 3.0 | 14.0 | | 6 | 5.3 |
| | R | 9.2 | 21.0 | 9.8 | 21.0 | 9.0 | 21.0 | | 4.5 | 4.5 |
| User Mix 3GB | S | 1.8 | 6.0 | 2.0 | 7.5 | 2.6 | 9.0 | | 6 | 5.3 |
| | R | 9.5 | 29.0 | 9.5 | 29.0 | 9.5 | 29.0 | | 14 | 11.5 |
| User Mix 12GB | S | | | | | | | | | |
| | R | | | | | | | | | |
| Large File 4GB | S | 1.8 | 6.0 | 2.0 | 7.2 | 2.7 | 9.0 | | 6 | 5.6 |
| | R | 9.7 | 31.0 | 9.7 | 31.0 | 9.7 | 31.0 | | 21 | 16.5 |
| Large File 32GB | S | | | | | | | | | |
| | R | | | | | | | | | |
| Large File 64GB | S | | | | | | | | | |
| | R | | | | | | | | | |
| 1 Directory Many Objects | S | 1.8 | 1.8 | 2.2 | 2.2 | 2.6 | 2.6 | | | |
| | R | 7.5 | 7.5 | 7.7 | 7.7 | 7.8 | 7.7 | | | |
| Many Directories Many Objects | S | 1.8 | 1.8 | 2.2 | 2.2 | 2.6 | 2.6 | | | |
| | R | 5.4 | 5.4 | 6.0 | 6.0 | 6.0 | 6.0 | | | |
| Domino Mail Files | S | 1.8 | 1.8 | 2.0 | 2.0 | 2.6 | 2.6 | | | |
| | R | 9.6 | 9.6 | 9.8 | 9.8 | 9.8 | 9.8 | | | |
| Network Storage Space | S | 1.8 | 1.8 | 2.0 | 2.0 | 2.6 | 2.6 | | | |
| | R | 9.6 | 9.6 | 9.8 | 9.8 | 9.8 | 9.8 | | | |

## 15.22 Software Compression

The rates a customer will achieve will depend upon the system resources available. This test was run in a very favorable environment to try to achieve the maximum rates. Software compression rates were gathered using the QSRSAVO API. The CPU used in all compression schemes was near 100%. The compression algorithm cannot span CPUs so the fact that measurements were performed on a 24-way system doesn't affect the software compression scenario.

*Table 15.22.1 - Measurements on an 840 24-way system 1080 RAID protected DASD Units (GB/HR) 128 GB mainstore*

| | | NSRC1GB | NUMX12GB | SR16GB | Software Compression Ratio |
|---|---|---|---|---|---|
| iV5R1 | Save | 19 | 135 | 170 | |
| | Restore | 7 | 45 | 170 | |
| iV5R2 | Save | 19 | 200 | 480 | |
| | Restore | 7 | 50 | 480 | |
| iV5R2 Using API DTACPR *LOW | Save | | 88 | 108 | 1.5:1 |
| | Restore | | 37 | 57 | |
| iV5R2 Using API DTACPR *MED | Save | | 26 | 27 | 2.7:1 |
| | Restore | | 23 | 31 | |
| iV5R2 Using API DTACPR *HIGH | Save | | 6 | 6 | 3:1 |
| | Restore | | 39 | 65 | |

## 15.23 9406-MMA DVD RAM

*Table 15.23.1 - iV5R4M5 Measurements on an 9406-MMA 4-way system 6 Mirrored DASD in the CEC and 24 RAID5 protected DASD Units attached 32 GB memory Measurements in (GB/HR) all 30 DASD in the system ASP.*

| Workload<br>S = Save<br>R = Restore | | SAS 6331 DTACPR *NO 5X Media | SAS 6331 DTACPR *YES 5X Media | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Release Measurements were done | | iV5R4M5 | iV5R4M5 | | | | | | | |
| Source File 1GB | S | 3.0 | 13.4 | | | | | | | |
| | R | 7.3 | 9.3 | | | | | | | |
| User Mix 3GB | S | 2.3 | 8.0 | | | | | | | |
| | R | 12.5 | 28.0 | | | | | | | |
| Large File 4GB | S | 2.2 | 8.0 | | | | | | | |
| | R | 14.0 | 45.0 | | | | | | | |
| 1 Directory<br>Many Objects | S | 2.3 | 2.3 | | | | | | | |
| | R | 9.0 | 9.0 | | | | | | | |
| Many Directories<br>Many Objects | S | 2.2 | 2.2 | | | | | | | |
| | R | 5.5 | 5.5 | | | | | | | |
| Domino Mail Files | S | 2.3 | 2.3 | | | | | | | |
| | R | 14.5 | 14.5 | | | | | | | |
| Network Storage Space | S | 2.2 | 2.2 | | | | | | | |
| | R | 14.0 | 14.0 | | | | | | | |

## 15.24 9406-MMA 576B IOPLess IOA

*Table 15.24.1 - iV6R1M0 Measurements on an 9406-MMA 4-way system 200 RAID5 protected DASD Units in the system ASP, attached via 571F IOAs 40 GB memory Measurements in (GB/HR). Two different Virtual tape experiments with 60 RAID5 DASD ASP2 and 120 RAID5 DASD in ASP2*

| Workload S = Save R = Restore | | 3580 Ultrium 3 576B 2 Port 4Gb Fiber Adapter | 3580 Ultrium 4 576B 2 Port 4Gb Fiber Adapter | 3592E02 Fiber 576B 2 Port 4Gb Fiber Adapter | 5746 Half High Ultrium 4 572A Adapter | 3592E06 Fiber 576B 2 Port 4Gb Fiber Adapter | Virtual Tape 60 DASD in ASP2 | Virtual Tape 120 DASD in ASP2 | Two High Speed Tape Drives on a Single 576B IOA using both ports concurrently | |
|---|---|---|---|---|---|---|---|---|---|---|
| IBM i Release | | V6R1M0 | V6R1M0 | V6R1M0 | V6R1M0 | V6R1M0 | V6R1M0 | V6R1M0 | V6R1M0 | |
| Source File 1GB | S | 40 | 32 | 34 | 34 | 34 | 40 | 40 | | |
| | R | 45 | 50 | 50 | 50 | 50 | 42 | 42 | | |
| User Mix 12GB | S | 280 | 234 | 230 | 230 | 230 | 220 | 280 | | |
| | R | 190 | 210 | 230 | 210 | 230 | 220 | 220 | | |
| Large File 64GB | S | 615 | 859 | 885 | 700 | 1050 | 350 | 770 | | |
| | R | 590 | 837 | 810 | 700 | 1000 | 750 | 770 | 1st Drive | 2nd Drive |
| Large File 320GB | S | 625 | 890 | 920 | 700 | 1100 | 350 | 770 | 920 | 885 |
| | R | 590 | 890 | 845 | 700 | 1000 | 750 | 770 | 485 | 475 |
| 1 Directory Many Objects | S | 50 | 55 | 55 | 55 | 55 | 50 | 50 | | |
| | R | 50 | 50 | 50 | 50 | 50 | 50 | 50 | | |
| Many Directories Many Objects | S | 40 | 40 | 40 | 40 | 40 | 38 | 38 | | |
| | R | 26 | 28 | 27 | 27 | 27 | 26 | 26 | | |
| Domino Mail Files | S | 450 | 575 | 580 | 550 | 650 | 330 | 700 | | |
| | R | 450 | 650 | 650 | 650 | 750 | 700 | 700 | | |

## 15.25 What's New and Tips on Performance

### What's New

**iV6R1M0**
**March 2008**
BRMS-Based Save/Restore Software Encryption and DASD-Based ASP Encryption
576B IOPLess Storage IOA

**iV5R4M5**
**July 2007**
3580 Ultrium 4 - 4Gb Fiber Channel Tape Drive
6331 SAS DVD RAM for 9406-MMA system models

**iV5R4**
**January 2007**
571E and 571F storage IOAs (see DASD Performance chapter for more information)

**August 2006**
1. DUPTAP performance PTFs (iV5R4 - SI23903, MF39598, MF39600, and MF39601)
2. 3580 Ultrium 3 4Gb Fiber Channel Tape Drive

**January 2006**
1. Virtual Tape
2. Parallel IFS
3. 3580 Ultrium 3 2Gb Fiber Channel Tape Drive
4. 3592E
5. VXA-320
6. ½ High Ultrium 2
7. IFS Restore Improvements for the Directory Workloads
8. 5761 4Gb Fiber Adapter

### TIPS

1. Backup devices are affected by the media type. For most backup devices the right media and density can greatly affect the capacity and speed of your save or restore operation. **USE THE RIGHT MEDIA FOR YOUR BACKUP DEVICE.** (i.e. Use a 25 GB tape cartridge in a 25 GB drive).
2. A Backup and Recovery Management System such as BRMS/400 is recommended to keep track of the data and make the most of multiple backup devices.
3. Domino Online Performance Tips:

Http://www-03.ibm.com/servers/eserver/iseries/service/brms/domperftune.html

# Chapter 16 IPL Performance

Performance information for Initial Program Load (IPL) is included in this section.

The primary focus of this section is to present observations from IPL tests on different System i models. The data for both normal and abnormal IPLs are broken down into phases, making it easier to see the detail. For information on previous models see a prior Performance Capabilities Reference.

**NOTE**: The information that follows is based on performance measurements and analysis done in the Server Group Division laboratory. Actual performance may vary significantly from these tests.

## 16.1 IPL Performance Considerations

The wide variety of hardware configurations and software environments available make it difficult to characterize a 'typical' IPL environment and predict the results. The following section provides a simple description of the IPL tests.

## 16.2 IPL Test Description

Normal IPL

- Power On IPL (cold start after managed system was powered down completely)
- For a normal IPL, benchmark time is measured from power-on until the System i server console sign-on screen is available.

Abnormal IPL

- System abnormally terminated causing recovery processing to be done during the IPL. The amount of processing is determined by the system activities at the time the system terminates.
- For an abnormal IPL, the benchmark consists of bringing up a database workload and letting it run until the desired number of jobs are running on the system. Once the workload is stabilized, the system is forced to terminate, forcing a mainstore dump (MSD). The dump is then copied to DASD via the Auto Copy function. The Auto Copy function is enabled through System Service Tools (SST). The System i partition is set to normal so that once the dump is copied, the system completes the remaining IPL with no user intervention. Benchmark time is measured from the time the system is forced to terminate, to when the System i server console sign on screen is available.
- Settings: on the CHGIPLA command the parameter, HDWDIAG, set to (*MIN). All physical files are explicitly journaled. Also logical files are journaled using SMAPP (System Managed Access Path Protection) by using the EDTRCYAP command set to *MIN.

**NOTE**: Due to some longer starting tasks ( like TCP/IP ), all workstations may not be up and ready at the same time as the console workstation displays a sign-on screen.

## 16.3 9406-MMA System Hardware Information

### 16.3.1 Small system Hardware Configuration

9406-MMA 7051 4 way - 32 GB Mainstore
DASD / 30 70GB 15K rpm arms,
6 DASD in CEC Mirrored
24 DASD in a #5786 EXP24 Disk Drawer attached with a 571F IOA RAID5 Protected

**Software Configuration**

100,000 spool files (100,000 completed jobs with 1 spool file per job)
500 jobs in job queues (inactive)
600 active jobs in system during Mainstore dump
1000 user profiles, 1000 libraries
**Active Database:** 2 libraries with 500 physical files and 20 logical files

### 16.3.2 Large system Hardware Configurations

9406-MMA 7056 8 way - 96 GB Mainstore
DASD / 432 70GB 15K rpm arms,
3 ASP's defined, 108 RAID5 DASD in ASP1, 288 RAID5 DASD in ASP2, 36 DASD no protection in ASP3 - Mainstore dump set to ASP 2

- This system was tested with database files unrelated to this test covering 30% of the DASD space available, this database load causes a long directory recovery.

9406-MMA 7061 16 way - 512 GB Mainstore
DASD / 1000 70GB 15K rpm arms,
3 ASP's defined, 196 Nonconfigured DASD, 120 RAID5 DASD in ASP1, 612 RAID5 DASD in ASP2, 72 DASD no protection in ASP3 - Mainstore dump set to ASP 2

- This system was tested with database files unrelated to this test covering 30% of the DASD space available, this database load causes a long directory recovery.

**Software Configuration**

400,000 spool files (400,000 completed jobs with 1 spool files each)
1000 jobs waiting on job queues (inactive)
11000 active jobs in system during mainstore dump
200 remote printers, 6000 user profiles, 6000 libraries
**Active Database:**

- 25 libraries with 2600 physical files and 452 logical files
- 2 libraries with 10,000 physical files and 200 logical files

NOTE:

- Physical files are explicitly journaled
- Logical files are journaled using SMAPP set to *MIN
- Commitment Control used on 20% of the files

## 16.4 9406-MMA IPL Performance Measurements (Normal)

The following tables provide a comparison summary of the measured performance data for a normal and abnormal IPL. Results provided do not represent any particular customer environment.

Measurement units are in minutes and seconds

*Table 16.4.1 iV5R4M5 Normal IPL - Power-On (Cold Start)*

| | iV5R4M5<br>GA1 Firmware<br>4 Way<br>9406-MMA<br>7051<br>32 GB<br>30 DASD | iV6R1<br>GA3 Firmware<br>4 Way<br>9406-MMA<br>7051<br>32 GB<br>30 DASD | iV5R4M5<br>GA1 Firmware<br>8 Way<br>9406-MMA<br>7056<br>96 GB<br>432 DASD | iV5R4M5<br>GA1 Firmware<br>16 Way<br>9406-MMA<br>7061<br>512 GB<br>1000 DASD | iV6R1<br>GA3 Firmware<br>16 Way<br>9406-MMA<br>7061<br>512 GB<br>1000 DASD |
|---|---|---|---|---|---|
| **Hardware** | **3:10** | **3:12** | **7:53** | **19:17** | **22:07** |
| **SLIC** | **4:49** | **5:07** | **7:53** | **10:05** | **9:58** |
| **OS/400** | **:48** | **1:23** | **2:12** | **2:41** | **2:22** |
| **Total** | **8:47** | **9:42** | **17:58** | **32:03** | **34:27** |

Generally, the hardware phase is composed of C1xx xxxx, C3xx xxxx and C7xx xxxx. SLIC is composed of C200 xxxx and C600 xxxx. OS/400 is composed of C900 xxxx SRCs to the System i server console sign-on.

## 16.5 9406-MMA IPL Performance Measurements (Abnormal)

Measurement units are in minutes and seconds.

*Table 16.5.1 iV5R4M5 Abnormal IPL (Partition MSD)*

| | iV5R4M5<br>GA1 Firmware<br>4 Way<br>9406-MMA<br>7051<br>32 GB<br>30 DASD | iV6R1<br>GA3 Firmware<br>4 Way<br>9406-MMA<br>7051<br>32 GB<br>30 DASD | iV5R4M5<br>GA1 Firmware<br>8 Way<br>9406-MMA<br>7056<br>96 GB<br>432 DASD | iV5R4M5<br>GA1 Firmware<br>16 Way<br>9406-MMA<br>7061<br>512 GB<br>1000 DASD | iV6R1<br>GA3 Firmware<br>16 Way<br>9406-MMA<br>7061<br>512 GB<br>1000 DASD |
|---|---|---|---|---|---|
| **Processor MSD** | **1:50** | **1:02** | **4:12** | **4:28** | **4:34** |
| **SLIC MSD IPL with Copy** | **7:23** | **10:45** | **7:00** | **11:35** | **10:56** |
| **Shutdown re-ipl** | **2:00** | **2:24** | **3:18** | **2:28** | **3:04** |
| **SLIC re-ipl** | **3:09** | **1:29** | **2:32** | **4:02** | **3:28** |
| **OS/400** | **4:22** | **5:04** | **28:06** | **29:27** | **20:47** |
| **Total** | **18:44** | **20:44** | **45:08** | **52:00** | **42:49** |

## 16.6 NOTES on MSD

MSD is Mainstore Dump. General IPL phase as it relates to the SRCs posted on the operation panel: Processor MSD includes the D2xx xxxx and C2xx xxxx right after the system is forced to terminate. SLIC MSD IPL with Copy follows with the next series of C6xx xxxx, see the next heading for more information on the SLIC MSD IPL with Copy. The copy occurs during the C6xx 4404 SRCs. Shutdown includes the Dxxx xxxx SRCs. Hardware re-ipl includes the next phase of D2xx xxxx and C2xx xxxx. SLIC re-IPL follows which are the C600 xxxx SRCs. OS/400 completes with the C900 xxxx SRCs.

### 16.6.1 MSD Affects on IPL Performance Measurements

SLIC MSD IPL with Copy is affected by the number of DASD units and the jobs executing at the time of the mainstore dump.

When a system is abnormally terminated, in-process changes to the directories used by the system to manage storage may be lost. During the subsequent IPL, storage management directory recovery is performed to ensure the integrity of the directories and the underlying storage allocations.

The duration of this recovery step will depend on the type of recovery performed and on the size of the directories. In most cases, a subset directory recovery (SRC C6004250) will be performed which may typically run from 2 minutes to 30 minutes depending upon the system. In rare cases, a full directory recovery (SRC C6004260) is performed which typically runs much longer than a subset directory recovery. The duration of the subset directory recovery is dependent on the size of the directory (which relates to the amount of data stored on the system) and on the amount of in-process changes. With the amount of data stored on our largest configurations with one to two thousand disk units, subset directory recovery (SRC C6004250) took from 14 minutes to 50 minutes depending upon the system.

**DASD Unit's Effect on MSD Time** - Through experimental testing we found the time spent in MSD copying the data to disk is related to the number of DASD arms available. Assigning the MSD copy to an ASP with a larger number of DASD can help reduce your recovery time if an MSD should occur.

## 16.7 5XX System Hardware Information

### 16.7.1 5XX Small system Hardware Configuration

520 7457 2 way - 16 GB Mainstore
DASD / 23 35GB 15K rpm arms,
RAID Protected

**Software Configuration**
100,000 spool files (100,000 completed jobs with 1 spool file per job)
500 jobs in job queues (inactive)
500 active jobs in system during Mainstore dump
1000 user profiles
1000 libraries

**Database:**
- 2 libraries with 500 physical files and 20 logical files

### 16.7.2 5XX Large system Hardware Configuration

570 7476 16 way - 256 GB Mainstore
DASD / 924 35GB arms 15K rpm arms,
RAID protected, 3 ASP's defined, majority of the DASD in ASP2 - Mainstore dump was to ASP 2
- This system was tested with 2 TB of database files unrelated to this test, but this load causes a long directory recovery.

595 7499 32-way - 384 GB Mainstore
DASD / 1125 35GB arms 15K rpm arms
RAID protected, 3 ASP's defined, majority of the DASD in ASP2 - Mainstore dump was to ASP 2
- This system was tested with 4 TB of database files unrelated to this test, but this load causes a long directory recovery.

**Software Configuration**
400,000 spool files (400,000 completed jobs with 1 spool files each)
1000 jobs waiting on job queues (inactive)
11000 active jobs in system during mainstore dump
200 remote printers
6000 user profiles
6000 libraries

**Database:**
- 25 libraries with 2600 physical files and 452 logical files
- 2 libraries with 10,000 physical files and 200 logical files

NOTE:
- Physical files are explicitly journaled
- Logical files are journaled using SMAPP set to *MIN
- Commitment Control used on 20% of the files

## 16.8 5XX IPL Performance Measurements (Normal)

The following tables provide a comparison summary of the measured performance data for a normal and abnormal IPL. Results provided do not represent any particular customer environment.

Measurement units are in minutes and seconds

*Table 16.8.1 Normal IPL - Power-On (Cold Start)*

| | V5R3 GA3 Firmware 2 Way 520 7457 16 GB 23 DASD | iV5R4 GA7 Firmware 2 Way 520 7457 16 GB 23 DASD | V5R3 GA3 Firmware 16 Way 570 7476 256 GB 924 DASD | iV5R4 GA7 Firmware 16 Way 570 7476 256 GB 924 DASD | V5R3 GA3 Firmware 32 Way 595 7499 384 GB MS 1125 DASD | iV5R4 GA7 Firmware 32 Way 595 7499 384 GB 1125 DASD |
|---|---|---|---|---|---|---|
| **Hardware** | 5:19 | 3:30 | 18:37 | 17:44 | 25:50 | 26:27 |
| **SLIC** | 3:49 | 4:30 | 6:42 | 6:43 | 8:50 | 9:36 |
| **OS/400** | 1:00 | :50 | 1:32 | 2:32 | 2:30 | 3:43 |
| **Total** | 10:08 | 8:50 | 26:51 | 26:59 | 37:10 | 39:46 |

The workloads were increased for iV5R4 to better reflect common system load affecting the OS/400 portion of the IPL

Generally, the hardware phase is composed of C1xx xxxx, C3xx xxxx and C7xx xxxx on the 5xx systems. SLIC is composed of C200 xxxx and C600 xxxx. OS/400 is composed of C900 xxxx SRCs to the IBM i operating system console sign-on.

## 16.9 5XX IPL Performance Measurements (Abnormal)

Measurement units are in hours, minutes and seconds.

*Table 16.9.1 Abnormal IPL (Partition MSD)*

| | V5R3 GA3 Firmware 2 Way 520 7457 16 GB 23 DASD | iV5R4 GA7 Firmware 2 Way 520 7457 16 GB 23 DASD | V5R3 GA3 Firmware 16 Way 570 7476 256 GB 924 DASD | iV5R4 GA7 Firmware 16 Way 570 7476 256 GB 924 DASD | V5R3 GA3 Firmware 32 Way 595 7499 384 GB MS 1125 DASD | iV5R4 GA7 Firmware 32 Way 595 7499 384 GB 1125 DASD |
|---|---|---|---|---|---|---|
| **Processor MSD** | 00:35 | 4:54 | 01:53 | 6:39 | 02:41 | 6:06 |
| **SLIC MSD IPL with Copy** | 04:50 | 15:40 | 24:10 | 42:18 | 43:10 | 40:03 |
| **Shutdown re-ipl** | 02:46 | 2:50 | 04:19 | 2:23 | 03:59 | 3:57 |
| **SLIC re-ipl** | 01:59 | 2:17 | 03:59 | 5:22 | 04:16 | 6:21 |
| **OS/400** | 03:21 | 4:20 | 09:56 | 25:45 | 13:56 | 44:10 |
| **Total** | 13:31 | 30:01 | 44:17 | 1:22:27 | 1:08:02 | 1:40:37 |

The workloads were increased for iV5R4 to better reflect common system load affecting the MSD and the OS/400 portion of the IPL

## 16.10 5XX IOP vs IOPLess effects on IPL Performance (Normal)

Measurement units are in minutes and seconds.

*Table 16.10.2 Normal IPL - Power-On (Cold Start)*

| | **iV5R4<br>GA7 Firmware<br>16 Way IOP<br>570<br>7476<br>256 GB<br>924 DASD** | **iV5R4<br>GA7 Firmware<br>16 Way IOPLess<br>570<br>7476<br>256 GB<br>924 DASD** |
|---|---|---|
| **Hardware** | 17:44 | 18:06 |
| **SLIC** | 6:43 | 7:20 |
| **OS/400** | 2:32 | 2:52 |
| **Total** | 26:59 | 28:18 |
| | | |

## 16.11 IPL Tips

Although IPL duration is highly dependent on hardware and software configuration, there are tasks that can be performed to reduce the amount of time required for the system to perform an IPL. The following is a partial list of recommendations for IPL performance:

- Remove unnecessary spool files. Use the Display Job Tables (DSPJOBTBL) command to monitor the size of the job table(s) on the system. Change IPL Attributes (CHGIPLA) command can be used to compress job tables if there is a large number of available job table entries. The IPL to compress the tables maybe longer, so try to plan it along with a normal maintenance IPL where you have the time to wait for the table to compress.

- Reduce the number of device descriptions by removing any obsolete device descriptions.

- Control the level of hardware diagnostics by setting the CHGIPLA command to specify HDWDIAG(*MIN), the system will perform only a minimum, critical set of hardware diagnostics. This type of IPL is appropriate in most cases. The exceptions include a suspected hardware problem, or when new hardware, such as additional memory, is being introduced to the system.

- Reduce the amount of rebuild time for access paths during an IPL by using System Managed Access Path Protection (SMAPP). The IBM i operating system Backup and Recovery book (SC41-5304) describes this method for protecting access paths from long recovery times during an IPL.

- For additional information on how to improve IPL performance, refer to *IBM i operating system Basic System Operation, Administration, and Problem Handling (SC41-5206)* - or to the redbook *The System Administrator's Companion to IBM i operating system Availability and Recovery (SG24-2161).*

# Chapter 17. Integrated BladeCenter and System x Performance

*This chapter provides a performance overview and recommendations for the Integrated xSeries Server[4], the Integrated xSeries Adapter and the iSCSI host bus adapter. In addition, the chapter presents some performance characteristics and impacts of these solutions on System i™.*

## 17.1 Introduction

The Internet SCSI Host Bus Adapter (iSCSI HBA), the Integrated xSeries® Server for iSeries™ (IXS), and the Integrated xSeries Adapter (IXA) extend the utility of the System i solution by integrating x86 and AMD based servers with the System i platform. Selected models of Intel based servers may run Windows® 2000 Server editions, Windows Server 2003 editions, Red Hat® Enterprise Linux®, or SUSE® LINUX Enterprise Server. In addition, the iSCSI HBA allows System i models to integrate and control IBM System x and IBM BladeCenter® model servers.

For more information about supported models, operating systems, and options, please see the "System i integration with BladeCenter and System x" web page referenced at the end of this chapter. Also, see the iSeries Information Center content titled "Integrated operating environments - Windows environment on iSeries" for iSCSI and IXS/IXA concepts and operation details.

In the text following, the System i platform is often referred to as the "host" server. The IXS, IXA attached server, or the iSCSI HBA attached server is referred to as the "guest" server.

**V5R4 iSCSI Host Bus Adapter** (iSCSI HBA)

The iSCSI host bus adapters (hardware type #573B Copper and #573C Fiber-optic) have been introduced in V5R4. The iSCSI HBA supports 1Gbit Ethernet network connections, and provides i5/OS system management and disk consolidation for guest System x and BladeCenter platforms.

The iSCSI HBA solution provides an extensive scalability range - from connecting up to 8 guest servers through one iSCSI HBA for a lower cost connectivity, to allowing up to 4 iSCSI HBAs per individual guest server for scalable bandwidth. For information about the numbers of supported adapters please see the "System i integration with BladeCenter and System x" web page.

With the iSCSI HBA solution, no disks are installed on the guest servers. The host i5/OS server provides disks, storage consolidation, guest server management, along with the tape, optical, and virtual ethernet devices. Currently, only Windows 2003 Server editions, with SP1 or release 2, are supported with the iSCSI solution.

**The Integrated xSeries Adapter** (IXA)

The Integrated xSeries Adapter (IXA - hardware type #2689-001 or #2689-002) is a PCI-based interface card that installs inside selected models of System x, providing a High Speed Link (HSL) connection to a host i5/OS system. The guest server provides the processors, memory, and Server Proven adapters, but no disks[5]. IXA attached SMP servers support larger workloads, more users and greater flexibility to attach devices than the IXS uni-processor models.

[4] The IBM System i and IBM System x product family names have replaced the IBM eSeries iSeries and xSeries product family names. However, the IXS and IXA adapters retain the iSeries and xSeries brand labels.

[5] With the IXA - all the disk drives used by the guest server are under the control of the host server. The are no disk drives in the guest server.

**Integrated xSeries Servers** (IXS)

An Integrated xSeries Server is an Intel processor-based server on a PCI-based interface card that plugs into a host system. This card provides the processor, memory, USB interfaces, and in some cases, a built-in gigabit Ethernet adapter. There are several hardware versions of the IXS:

- The 2.0 GHz Pentium® M IXS (hardware type #4812-001)[6].
- The 2.0 GHz PCI IXS (hardware type #2892-002).

Older versions of the IXS Card are: the 1.6 GHz PCI IXS (hardware type #2892-001), 1 GHz (type #2890-003), 850 Mhz (type #2890-002), and 700 MHz (type #2890-001).

## 17.2 Effects of Windows and Linux loads on the host system

The impact of IXS and IXA device I/O operations on the host system is similar for all versions of the IXA and IXS cards. The IXA and IXS cards and drivers channel the host directed device I/O through an Input / Output processor (IOP) on the card..

The iSCSI HBA is an IOP-less input/output adapter. The internal operation and performance characteristics differ from the IXS/IXA solutions - with a slightly higher host CPU, and memory requirements. However, the iSCSI solution offers greater scalability and overall improved performance compared to the IXS/IXA. The iSCSI solution adds support for IBM BladeCenter and additional System x models.

Depending on the Windows or Linux application activity, integrated guest server I/O operations impose an indirect load on the System i native CPU, memory and storage subsystems. The rest of this chapter describes some of the performance and memory resource impacts.

### 17.2.1 IXS/IXA Disk I/O Operations:

The integrated xSeries servers use i5/OS network server storage spaces for its hard drives, which are allocated out of i5/OS storage.

For the IXS/IXA server - IBM supplied virtual SCSI drivers render these storage spaces as physical disks in Windows or Linux. The device drivers cooperate with i5/OS to perform the disk operations, and additional host CPU resource is used during disk access, along with a fixed amount of storage. The amount of CPU impact is a function of the disk I/O rate and disk operation size.

The disk linkage type and Windows driver write cache property alters the CPU cost and average write latency.

- Fixed and Dynamically Linked Disks

  Fixed (statically) and dynamically linked disk drives have the same performance characteristics. One advantage of dynamically linked drives is that in most cases, they may be linked and unlinked while the server is active.

- Shared Disks

  System i Windows integration supports the Microsoft Clustering Service, which supports “shared” disks. When a storage space is linked as a “shared” disk, all write operations require extended communications to insure data integrity, which slightly increases the host CPU cost and response time.

[6] Requires a separate IOP card, which is included with features 4811, 4812 and 4813.

- Write Cache Property

  When the disk device write cache property is disabled, disk operations have similar performance characteristics to shared disks. You may examine or change the “Write Cache” property on Windows by selecting disk “properties” and then the “Hardware tab”. Then view “Properties” for a selected disk and view the “Disk Properties” or “Device Options” tab.

  All dynamically and statically linked storage spaces have “Write Cache” enabled by default. Shared links have “Write Cache” disabled by default. While it is possible to enable “Write Cache” on shared disks, we recommend to keep it disabled to insure integrity during clustering fail-over operations. There is also negligible performance benefit to enabling the write cache on shared disks.

- Extended Write Operations

  Even though a Windows disk driver may have write cache enabled, the system or applications consider some write operations sensitive enough to request extended writes or flush operations “write through” to the disk. These operations incur the higher CPW cost regardless of the write caching property.

- For the IXS and IXA solutions - do not enable the disk driver “Enable advanced performance” property provided in Windows 2003. When enabled, all extended writes are turned into normal cached operations and flush operations are masked. This option is only intended to be used when the integrity of the write operations can be guaranteed via write through or battery backed memory. The IXS/IXA with write caching enabled cannot make this guarantee.

- IXS/IXA Disk Capacity Considerations

  The level of disk I/O achieved on the IXS or IXA varies depending on many variables, but given an adequate storage subsystem, the upper cap on I/O for a single server is limited by the IXS/IXA IOP component. Except in extreme test loads, it’s unlikely the IOP will saturate due to disk activity.

  When multiple IXS/IXA servers are attached under the same System i partition, the partition software imposes a cap on the aggregate total I/O from all the servers. It is not a strict limitation, but a typical capacity level is approximately 6000 to 10000 disk operations/sec.

### 17.2.2 iSCSI Disk I/O Operations:

- The iSCSI disk operations use a more scalable storage I/O access architecture than the IXS and IXA solutions. As a result, a single integrated server can scale to greater capacity by using multiple target and initiator iSCSI HBAs to allow multiple data paths.

- In addition, there is no inherent partition cap to the iSCSI disk I/O. The entire performance capacity of installed disks and disk IOAs is available to iSCSI attached servers.

- The Windows disk drive “write cache” policy does not directly affect iSCSI operations. Write operations always “write through” to the host disk IOAs, which may or may not cache in battery backed memory (depending on the capabilities and configuration of the disk IOA).

- iSCSI attached servers use non-reserved System i virtual storage in order to perform disk input or output. Thus, disk operations use host memory as an intermediate read cache. Write operations are flushed to disk immediately, but the disk data remains in memory and can be read on subsequent operations to the same sectors.

  While the disk operations page through a memory pool, the paging activity is not visible in the “Non-DB” pages counters displayable via the WRKSYSSTS command. This doesn’t mean the memory is not actively used, it’s just difficult to visualize how much memory is active. WRKSYSSTS will show faults and paging activity if the memory pool becomes constrained, but some write operations also result in faulting activity.

- With iSCSI, there are some Windows side disk configuration rules you must take into account to enable efficient disk operations. Windows disks should be configured as:
  - 1 disk partition per virtual drive.
  - File system formatted with cluster sizes of 4 kbyte or 4 kbyte multiples.
  - 2 gigabyte or larger storage spaces (for which Windows creates a default NTFS cluster size of 4kbytes).

  If necessary, you can use care to configure multiple disk partitions on a single virtual drive.
  - For storage spaces that are 1024 MB or less, make the partitions a multiple of 1 MB (1,048,576 bytes).
  - For storage spaces that are 511000 MB or less, the partition should be a multiple of 63 MB (66060288 bytes).
  - For storage spaces that are greater than 511000 MB, the partition should be a multiple of 252 MB (264,241,152 bytes).

  These guidelines allow file system structures to align efficiently between iSCSI Windows/Linux and i5/OS[7]. They allow i5/OS to efficiently manage the storage space memory, mitigate disk operation faulting activity, and thus improve overall iSCSI disk I/O performance. **Failure to follow these guidelines will cause iSCSI disk write operations to incur performance penalties**, including page faults and increased serialization of disk operations.

- In V5R4 – the CHGNWSSTG command and iSeries Navigator now supports the expansion of a storage space size. After the expansion, the file system in the disk should also be expanded - but take care on iSCSI disks: don't create a new partition in the expanded disk free space (unless the new partitions meeting the size guidelines above).

  The Windows 2003 "DISKPART" command can be used to perform the file system expansion – however it only actually expands the file system on "basic" disks. If a disk has been converted to a "dynamic" disk[8], the DISKPART command creates a new partition and configures a spanned set across the partitions. With iSCSI, the second partition may experience degraded disk performance.

### 17.2.3 iSCSI virtual I/O private memory pool

Applications sharing the same memory pool with iSCSI disk operations may be adversely impacted if the iSCSI network servers perform levels of disk I/O which can flush the memory pool. Thus, it is possible for other applications to begin to page fault because their memory has been flushed out to disk by the iSCSI operations. By default, the iSCSI virtual disk I/O operations occur through the *BASE memory pool.

In order to segregate iSCSI disk activity, V5R4 PTF SI23027 has been created to enable iSCSI virtual disk I/O operations to run out of an allocated private memory pool. The pool is enabled by creating a subsystem description named QGPL/QFPHIS and allocating a private memory pool of at least 4096 kilobytes. The amount of memory you want to allocate will depend on a number of factors, including number of iSCSI network servers, expected sustained disk activity for all servers, etc. See the "System i Memory Rules of Thumb" section below for more guidelines on the "iSCSI private pool" minimum size.

To activate the private memory pool for all iSCSI network servers, perform the following:

1. CRTSBSD SBSD(QGPL/QFPHIS) POOLS((1 10000 1))[9]

---

[7] These guidelines could also slightly improve IXS and IXA attached servers' disk performance, but to a much smaller degree.

[8] Not to be confused with "dynamically linked" storage spaces.

[9] Pick a larger or smaller pool size as appropriate. 10,000 KByte is a reasonable minimum value, but 4096 KByte is the absolute minimum supported.

2. Vary on any Network Server Description (NWSD) with a Network server connection type of *ISCSI.

During the iSCSI network server vary on processing the QFPHIS subsystem is automatically started if necessary. The subsystem will activate the private memory pool. iSCSI network server descriptions that are varied on will then utilize the first private memory pool configured with at least the minimum (4MB) size for virtual disk I/O operations.

The private memory pool is used by the server as long as the subsystem remains active. If the QFPHIS subsystem is ended prematurely (while an iSCSI network server is active), the server will continue to function properly but future virtual disk I/O operations will revert to the *BASE memory pool until the system memory pool is once again allocated.

**NOTE:** When ending the QFPHIS subsystem, i5/OS can reallocate the memory pool, possibly assigning the same identifier to another subsystem! Any active iSCSI network servers that are varied on and using the memory pool at the time the subsystem is ended may adversely impact other applications either when the memory pool reverts to *BASE or when the memory pool identifier is reassigned to another subsystem! To prevent unexpected impacts – do not end the QFPHIS subsystem while iSCSI servers are active.

### 17.2.4 Virtual Ethernet Connections:

The virtual Ethernet connections utilize the System i systems licensed internal code tasks during operation. When a virtual Ethernet port is used to communicate between Integrated servers, or between servers across i5/OS partitions, the host server CPU is used during the transfer. The amount of CPU used is primarily a function of the number of transactions and their size.

There are three forms of Virtual Ethernet connections used with the IXS/IXA and iSCSI attached servers:

- The “Point to point virtual Ethernet” is primarily used for the controlling partition to communicate with the integrated server. This network is called point to point because it has only two endpoints, the integrated server and the i5/OS platform. It is emulated within the host platform and no additional physical network adapters or cables are used. In host models, it is configured as an Ethernet line description with Port Number value *VRTETHPTP.
- A “Port-based”[10] virtual Ethernet connection allows IXS, IXA or iSCSI attached servers to communicate together over a virtual Ethernet (typically used for clustered IXS configurations), or to join an inter-LPAR virtual Ethernet available on non-POWER5 based systems This type of virtual Ethernet uses “network numbers”, and integrated servers can participate by configuring a port number value in the range *VRTETH0 through *VRTETH9.

  “Port-based” virtual Ethernet communications also require the host CPU to switch the communications data between guest servers.
- A “VLAN-based” (noted as Phyp in charts) virtual Ethernet connection allows IXS, IXA and iSCSI attached servers to participate in inter-LPAR virtual Ethernets. Each participating integrated server needs an Ethernet line description that associates a port value such as *VRTETH0 with a virtual adapter having a virtual LAN ID. You create the virtual adapter via the Hardware Management Console (HMC).

  VLAN-based communications also use the System i CPU to switch the communications data between server.

### 17.2.5 IXS/IXA IOP Resource:

---

[10] “Port-Based” refers to the original method of supporting VE introduced in V5R2 for models earlier than System i5. It is still available for integrated servers to communicate within a single partition on System i models.

IXS and IXA I/O operations (disk, tape, optical and virtual Ethernet) communications occur through the individual IXS and IXA IOP resource. This IOP imposes a finite capacity. The IOP processor utilization may be examined via the iSeries Collection Services utilities.

**The performance results presented in the rest of this chapter are based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here**.

## 17.3 System i memory rules of thumb for IXS/IXA and iSCSI attached servers.

The i5/OS machine pool memory "rule of thumb" is generally to size the machine pool with at least twice the active machine pool reserved size. Automatic performance adjustments may alter this according to the active load characteristics. But, there are base memory requirements needed to support the hardware and set of adapters used by the i5/OS partition. You can refer to the System i sales manual or Work Load Estimator for estimates of these base requirements. The "rules of thumb" below estimates the additional memory required to support iSCSI and IXS/IXA.

### 17.3.1 IXS and IXA attached servers:

I/O occurs through fixed memory in the machine pool. The IXS and IXA attached servers require approximately an additional 4MBytes of memory per server in the machine pool.

### 17.3.2 iSCSI attached servers:

IBM Director Server is required in each i5/OS partition running iSCSI HBA targets. IBM Director requires a minimum of 500 Mbytes in the base pool.

The specific memory requirements of iSCSI servers vary based on many configuration choices, including the number of LUNs, number of iSCSI target HBAs, number of NWSDs, etc. A suggested minimum memory "rule of thumb"[11] is:

[11] Based on a rough configuration of 5 LUNS per server, 2 VE connections per server, and two target HBA connections per server.

| | For Each Target HBA | For Each NWSD |
|---|---|---|
| Machine Pool: | 21 MBytes | 1 MByte |
| Base Pool: | 1 MByte | 0.5 MByte |
| QFPHIS Private Pool: | 0.5 MByte | 1 MByte[12] |
| Total: | 22.5 MBytes | 2.5 MBytes |

Warning: To ensure expected performance and continuing machine operation, it is critical to allocate sufficient memory to support all of the devices that are varied on. Inadequate memory pools can cause unexpected machine operation.

## 17.4 Disk I/O CPU Cost

| Disk Operation Rules of Thumb | CPWs[13] / 1k ops/sec |
|---|---|
| iSCSI linked disks | 190 |
| IXS/IXA static or dynamically linked disks with write caching enabled | 130 |
| IXS/IXA shared or quorum linked disks or write caching disabled | 155 |

While the disk I/O activity driven by the IXS/IXA or iSCSI is not strictly a "CPW" type load, the CPW estimate is still a useful metric to estimate the amount of i5/OS CPU required for a load. You can use the values above to estimate the CPW requirements if you know the expected I/O rate. For example, if you expect the Windows application server to generate 800 disk ops/sec on a dynamically or statically linked storage space, you can estimate the CPW usage as:

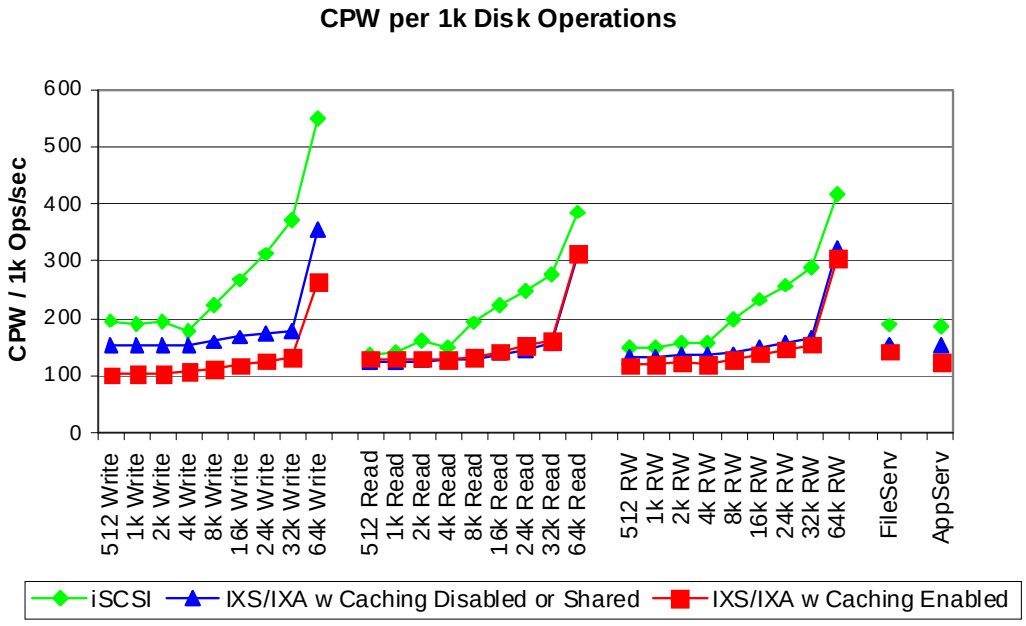
130 cpws/1kops * 800ops * 1kops/1000ops = 130 * 800/1000 = 104 CPWs

out of the host processor CPW capacity. While it is always better to project the performance of an application from measurements based on that same application, it is not always possible. This calculation technique gives a relative estimate of performance.

These rules of thumb are estimated from the results of performing file serving or application types of loads. In more detail, the chart below indicates an approximate amount of host processor (in CPW) required to perform a constant number of disk operations (1000) of various sizes. You can reasonably adjust this estimate linearly for your expected I/O level.

[12] Private pool assigned to QFPAIS must still be a 4 MB minimum size.

[13] A CPW is the "Relative System Performance Metric" from Appendix C. Note that the I/O CPU capacities may not scale exactly by rated system CPW, as the disk I/O doesn't represent a CPW type of load . This calculation is a convenient metric to size the load impacts. The measured CPW cost will actually decrease from the above values as the number of processors in the NWSD hosting partition increases, and may be higher than estimated when partial processors are used.




The charts shows the relative cost when performing 5 different types of operations[14].

- Random write operations of a uniform size (512, 1k, ... 64k).
- Random read operations of a uniform size (512, 1k, ... 64k).
- A 35% random write, 65% random read mix of operations with a uniform size (termed transaction processing type load).
- A fileserving type load - which consists of a mix of operations of various sizes similar in ratio to typical fileserving loads. This load is 80% random reads.
- An application server - database type load. This is also a mix to simulate the character of application and database type accesses - mostly random with about 40% reads.

### 17.4.1 Further notes about IXS/IXA Disk Operations

- Maximum disk operation size supported by the IXS or IXA is 32k. Thus, any Windows disk operations greater than 32k will result in the Windows operating system splitting the operation into 2 or more sequential operations.
- The IXS/IXA cost calculation is slightly greater on a POWER5® system (System i5) than on earlier 8xx systems. This includes some increased overhead costs in V5R4, and the new processor types. Use the newer rules of thumb listed above for CPW calculations.
- It does not matter if a storage space is linked statically or dynamically, the performance characteristics are identical.
- It does not matter if a server is an IXS or an IXA attached System x server, the disk performance is almost identical.

[14] Measured on a System i Model 570 - 2-way 26F2 processor (7495 capacity card), rated at 6350 CPWs, V5R4 release of i5/OS, 40 parity protected (RAID 5) 4326 disks, 3 2780 disk controllers. The IXA attached server was a x365xSeries (4way 2.5Ghz Xeon with IXA and Windows Server 2003 with SP1. The iSCSI servers were HS20 BladeCenter servers with a copper iSCSI (p/n 26K6489) daughter card. Switches were Nortel L2/3 Ethernet (p/n 26K6524) and Cisco Intelligent Gigabit Switch (p/n )

- A storage space which is linked as shared, or a disk with caching disabled, requires more CPU to process write operations (approx. 45%).
- Sequential operations cost approximately 10% less than the random I/O results shown above.
- Even though a Windows disk driver may have write cache enabled, some Windows applications may request to bypass the cache for some operations (extended writes), and these operations would incur the higher CPW cost.
- If your application load is skewed to the upper or lower average disk operations, you may encounter a smaller or larger CPW cost than indicated by the “rules of thumb”.

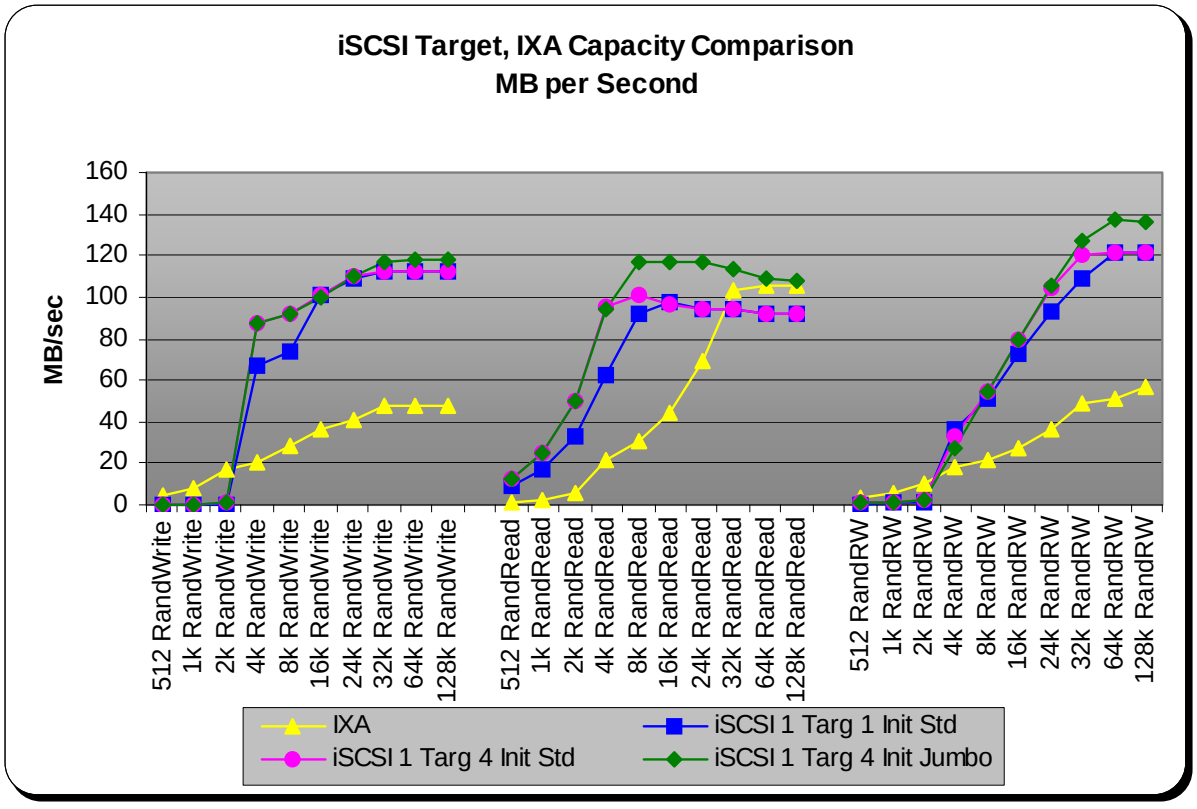
---

## 17.5 Disk I/O Throughput

The chart below compares the throughput performance characteristics of the current IXA product against the new iSCSI solution. The charts indicates an approximate capacity of a single target HBA when running various sizes and types of random operations.

The chart below also demonstrates that the new page based architecture utilized in the iSCSI solution provides better over all performance in Write and Read scenarios. While the sector based architecture of the current IXA provides slightly better performance in small block Write operations, that difference is quickly reversed when block sizes reach 4k. 4k block sizes and higher are representative of more I/O server scenarios.

As with all performance analysis, the actual values that you will achieve are dependent on a number of variables including workload, network traffic, etc..



The blue square line shows an iSCSI connection with a single target iSCSI HBA - single initiator iSCSI HBA connection, configured to run with standard frames. The pink circle line is a single target iSCSI HBA to multiple servers and initiators running also running with standard frames. With the initiators and switches configured to use 9k jumbo frames, a 15% to 20% increase in upper capacity is demonstrated.[15]

## 17.6 Virtual Ethernet CPU Cost and Capacities

If the virtual Ethernet connections are used for any significant LAN traffic, you need to account for additional System i CPU requirements. There is no single rule of thumb applicable to network traffics, as there are a great number of variables involved.

The charts below demonstrate approximate capacity for single TCP/IP connections, and illustrates the minimum CPW impacts for some network transaction sizes (send/receive operations) and types gathered with the Netperf [16]exerciser. The CPW chart below gives CPWs per Mbit/sec for increasing transaction sizes. When the transaction size is small, the CPW requirements are greater per transaction. When the arrival rate is high enough, some consolidation of operations within the process stream can occur and increase efficiency of operations.

Several charts are presented comparing the virtual ethernet capacity and costs between an iSCSI server running jumbo frames and standard frames, and a IXS or IXA server. In addition, a comparison of costs while using external NICs is added, to place the measurements in context. The "Point-to-Point" refers to the cost between an iSCSI, IXS or IXA attached server and an host system across the point to point connection. "Port Based VE" refers to a port-based connection between two guest servers in the same partition. "VLAN based VE" refers to a Virtual LAN based connection between two guest servers in the same partition, but using the VLAN to port associated virtual adapters. In the latter two cases, the total CPW cost would be split across partitions if the communication would occur between guest servers hosted by different partitions[17].

### 17.6.1 VE Capacity Comparisons

In general, VE has less capacity than an external Gigabit NIC. Greater capacity with VE is possible using 9k jumbo frames than with using standard 1.5k frames. Also, the iSCSI connection has a greater capacity

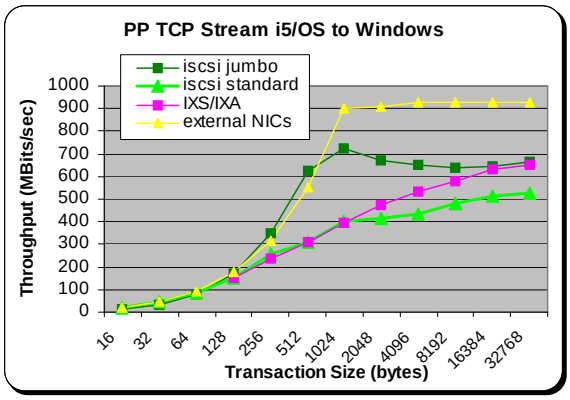
[15] In addition, jumbo frame configuration has no effect on the CPW cost of iSCSI disk operations.

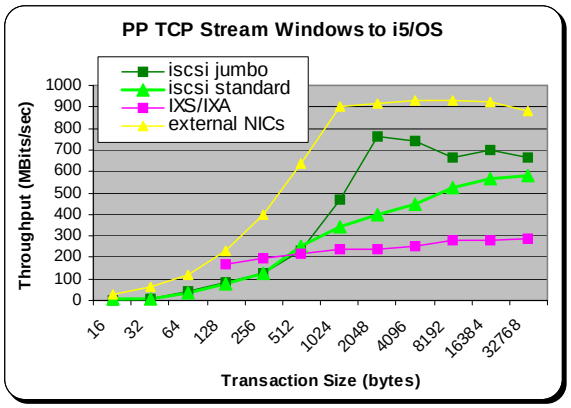
[16] Note that the Netperf benchmark consists of C programs which use a socket connection to read and write data between buffers. The CPW results above don't attempt to factor out the minimal application CPU cost. That is, the CPW results above include the primitive Netperf application, socket, TCP, and Ethernet operation costs. A real user application will only have this type of processing as a percentage of the overall workload.

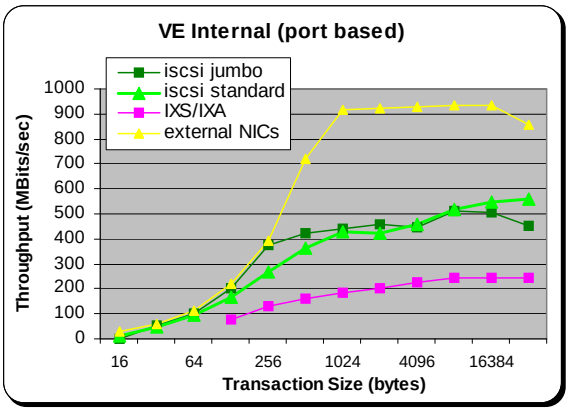
[17] Netperf TCP_STREAM measured on a System i Model 570 - 2-way 26F2 processor (7495 capacity card), rated at 6350 CPWs, V5R4 release of i5/OS. The IXA attached server was a x365xSeries (4way 2.5Ghz Xeon with IXA and Windows Server 2003 with SP1. The iSCSI servers were HS20 BladeCenter 32.Ghz uniprocessor servers with a copper iSCSI (p/n 26K6489) daughter card. Switches were Nortel L2/3 Ethernet (p/n 26K6524). This is only a rough indicator for capacity planning, actual results may differ for other hardware configurations.

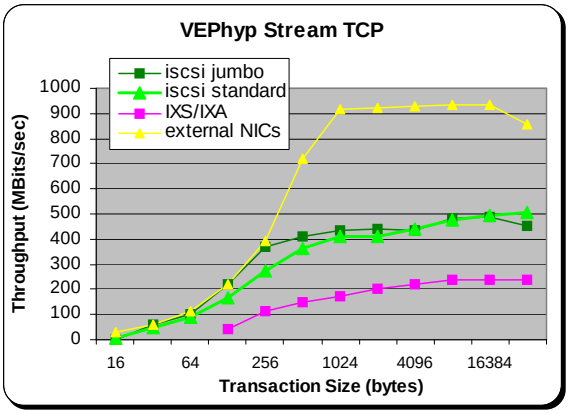

 

than an IXS or IXA attached VE connection. “Stream” means that the data is pushed in one direction, with only the TCP acknowledge packets running in the other direction.



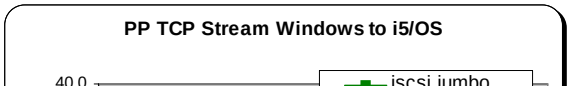



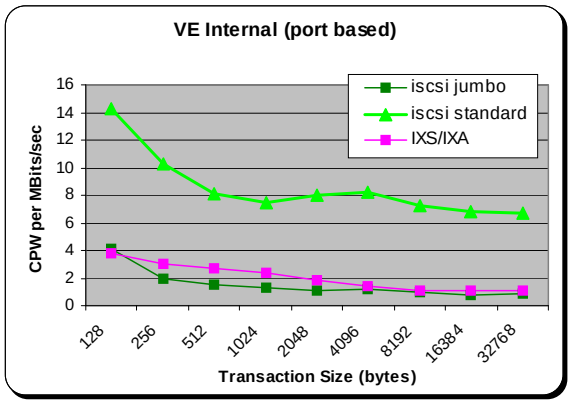



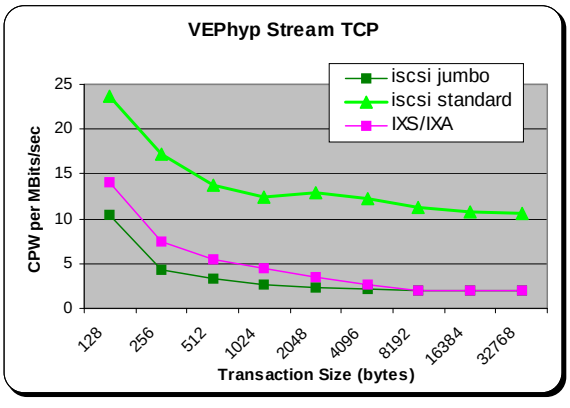




### 17.6.2 VE CPW Cost

CPW cost below is listed as CPW per Mbit/sec. For the point to point connection, the results are different depending on the direction of transfer. For connections between guest servers - the direction of transfer doesn’t matter to the results.

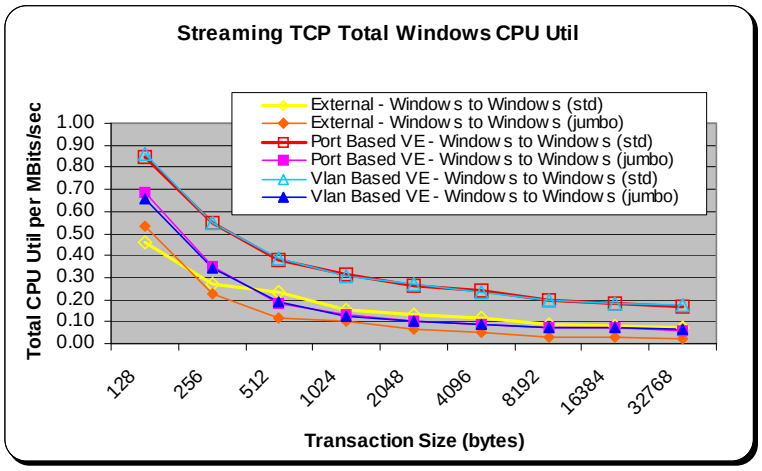
The chart above shows the CPW efficiency of operations (larger is better). Note the CPW per Mbits/sec scale on the left - as it's different for each chart.

For an IXS or IXA, the port-based VE has the least CPW or smaller packets due to consolidation of transfers available in Licensed Internal Code. The VLAN-based transfers have the greatest cost (However the total would be split during inter-LPAR communications).

For iSCSI, the cost of using standard frames is 1.5 to 4.5 times higher than jumbo frames.

### 17.6.3 Windows CPU Cost

The next chart illustrates the cost of iSCSI port-based, and VLAN-based virtual ethernet operations on a windows CPU. In this case, the CPUs used are 3.2Ghz Xeon uniprocessor between HS20 BladeCenter servers. This cost is compared to operations across the external gigabit NIC connections. Again, the jumbo frames operations are less expensive than standard frames, though the external NIC is twice as efficient in General.



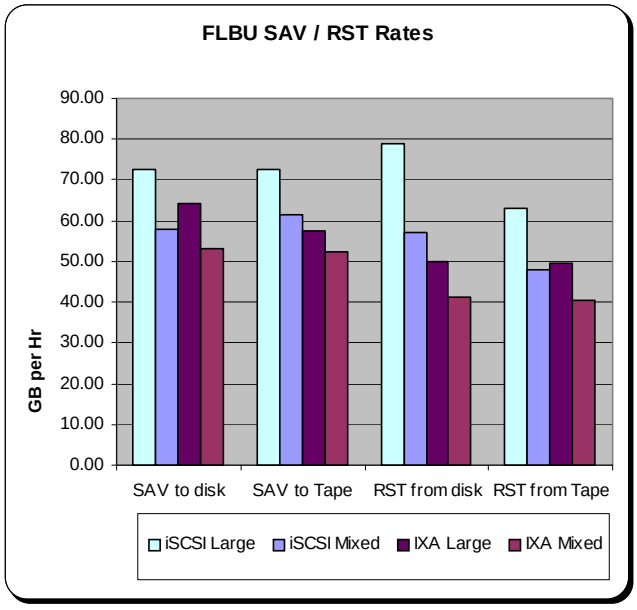

## 17.7 File Level Backup Performance

The Integrated Server support allows you to save integrated server data (files, directories, shares, and the Windows registry) to tape, optical or disk (*SAVF) in conjunction with your other i5/OS data. That is, this "file level backup" approach saves or restores the Windows files on an individual basis within the stream of other i5/OS data objects. It's not recommended that this approach is used as a primary backup procedure. Rather, you should still periodically save your NWS storage spaces and the NWSD associated with your Windows server or Linux server for disaster recovery.

Saving individual files does not operate as fast as saving the individual storage spaces. The save of a storage space on a equivalent machine and tape is about 210 Gbytes per hour, compared to the approximately 70 Gbytes per hour achieved using iSCSI below.

The chart below compares some SAV and RST rates for iSCSI and an IXA attached server. These results were measured on a System i Model 570 2-way 26F2 processor (7495 capacity card). The i5/OS release was V5R4. The IXA attached comparison server was an x365 xSeries (4way 2.5Ghz Xeon with IXA and Windows Server 2003 with SP1). The iSCSI servers were HS20 BladeCenter servers with a copper iSCSI (p/n 26K6489) daughter card. Switches were Nortel L2/3 Ethernet (p/n 26K6524). The target tape drive was a model 5755-001 (Ultrium LTO 2). All tests were run with jumbo frames enabled.

The legend label “Mixed Files” indicates a save of many files of mixed sizes - equivalent to the save of the Windows system file disk. “Large files” indicates a save of many large files - in this case many 100MB files.




## 17.8 Summary

The iSCSI host bus adapter, Integrated xSeries Server and Integrated xSeries Adapter provides scalable integration for full file, print and application servers running Windows 2000 Server, Windows Server 2003 or with Intel Linux editions. They provide flexible consolidation of System i solutions and Windows or Linux services, in combination with improved hardware control, availability, and reduced maintenance costs. These solutions perform well as a file or application server for popular applications, using the System i host disks, tape and optical resources. The iSCSI HBA addition in V5R4 increases Integrated server configuration flexibility and performance scalability. As part of the preparation for integrated server installations, care should be taken to estimate the expected workload of the Windows or Linux server applications and reserve sufficient i5/OS resources for the integrated servers.

## 17.9 Additional Sources of Information

***System i integration with BladeCenter and System x URL:***
http://www.ibm.com/systems/i/bladecenter/

***Redbook:*** “Microsoft Windows Server 2003 Integration with iSeries”, SG246959 at
http://www.redbooks.ibm.com/abstracts/SG246959.html

***Redbook:*** “Tuning IBM eServer xSeries Servers for Performance SG24-5287”
http://www.redbooks.ibm.com/abstracts/sg245287.html
While this document doesn’t address the integrated server configurations specifically, it is an excellent resource for understanding and addressing performance issues with Windows or Linux.

***Online documentation:*** “Integrated operating environments on iSeries”
http://publib.boulder.ibm.com/iseries/

Choose V5R4. In the “Contents” panel choose “iSeries Information Center”.
Expand “Integrated operating environments” and then “Windows environment on iSeries” for Windows environment information or “Linux” and then “Linux on an integrated xSeries solution for Linux Information on an IXS or attached xSeries server.

***Microsoft Hardware Compatibility Test URL:*** See
http://www.microsoft.com/whdc/hcl/search.mspx
search on IBM for product types Storage/SCSI Controller and System/Server Uniprocessor.

# Chapter 18. Logical Partitioning (LPAR)

## 18.1 Introduction

Logical partitioning (LPAR) is a mode of machine operation where multiple copies of operating systems run on a single physical machine.

A *logical partition* is a collection of machine resources that are capable of running an operating system. The resources include processors (and associated caches), main storage, and I/O devices. Partitions operate independently and are logically isolated from other partitions. Communication between partitions is achieved through I/O operations.

The *primary partition* provides functions on which all other partitions are dependent. Any partition that is not a primary partition is a *secondary partition*. A secondary partition can perform an IPL, can be powered off, can dump main storage, and can have PTFs applied independently of the other partitions on the physical machine. The primary partition may affect the secondary partitions when activities occur that cause the primary partition's operation to end. An example is when the PWRDWNSYS command is run on a primary partition. Without the primary partition's continued operation all secondary partitions are ended.

### V5R3 Information

Please refer to the whitepaper 'i5/OS LPAR Performance on POWER4 and POWER5 Systems' for the latest information on LPAR performance. It is located at the following website:

http://www-1.ibm.com/servers/eserver/iseries/perfmgmt/pdf/lparperf.pdf

### V5R2 Additions

In V5R2, some significant items may affect one's LPAR strategy (see "General Tips"):

- "Zero" interactive partitions. You do not have to allocate a minimum amount of interactive performance to every partition when V5R2 OS is in the Primary partition.

In V5R2, the customer no longer has to assign a minimum interactive percentage to LPAR partitions (can be 0). For partitions with no assigned interactive capability, LPAR system code will allow interactive as follows: 0.1% x (processors in partition/total processors) x processor CPW.

In V5R1, the customer had to allocate a minimum interactive percentage to LPAR partitions as follows: 1.5% x (processors in partition/total processors) x processor CPW. It is expected that the LPAR system code will issue a PTF to change the percentage from 1.5% to 0.5% for V5R1 systems.

Notes:
1. The above formulas yield the minimum ICPW for an LPAR region. The customer still has to divide this value by the total ICPW to get the percentage value to specify for the LPAR partition.
2. If there is not enough interactive CPW available for the partition given the previous formula ... the interactive percentage can be set to the percentage of the (processors in partition/total processors).

### General Tips

- Allocate fractional CPUs wisely. If your sizing indicates two partitions need 0.7 and 0.4 CPUs, see if there will be enough remaining capacity in one of the partitions with 0.6 and 0.4 or else 0.7 and 0.3 CPUs allocated. By adding fractional CPUs up to a "whole" processor, fewer physical processors will be used. Design implies that some performance will be gained.
- Avoid shared processors on large partitions if possible. Since there is a penalty for having shared processors (see later discussion), decide if this is really needed. On a 32 way machine, a whole processor is only about 3 per cent of the configuration. On a 24 way, this is about 4 per cent. Though we haven't measured this, the general penalty for invoking shared processors (often, five per cent) means that rounding up to whole processors may actually gain performance on large machines.

**V5R1 Additions**

In V5R1, LPAR provides additional support that includes: dynamic movement of resources without a system or partition reset, processor sharing, and creating a partition using Operations Navigator. For more information on these enhancements, click on System Management at URL: http://submit.boulder.ibm.com/pubs/html/as400/bld/v5r1/ic2924/index.htm

With processor sharing, processors no longer have to be dedicated to logical partitions. Instead, a shared processor pool can be defined which will facilitate sharing whole or partial processors among partitions. There is an additional system overhead of approximately 5% (CPU processing) to use processor sharing.

- Uniprocessor Shared Processors. You can now LPAR a single processor and allocate as little as 0.1 CPUs to a partition. This may be particularly useful for Linux (see Linux chapter).

---

## 18.2 Considerations

This section provides some guidelines to be used when sizing partitions versus stand-alone systems. The actual results measured on a partitioned system will vary greatly with the workloads used, relative sizes, and how each partition is utilized. For information about CPW values, refer to *Appendix D, "CPW, CIW and MCU Values for iSeries"*.

When comparing the performance of a standalone system against a single logical partition with similar machine resources, do not expect them to have identical performance values as there is LPAR overhead incurred in managing each partition. For example, consider the measurements we ran on a 4-way system using the standard AS/400 Commercial Processing Workload (CPW) as shown in the chart below.

For the standalone 4-way system we used we measured a CPW value of 1950. We then partitioned the standalone 4-way system into two 2-way partitions. When we added up the partitioned 2-way values as shown below we got a total CPW value of 2044. This is a 5% increase from our measured standalone 4-way CPW value of 1950. I.e. (2044-1950)/1950 = 5 %. The reason for this increased capacity can be attributed primarily to a reduction in the contention for operating system resources that exist on the standalone 4-way system.

Separately, when you compare the CPW values of a standalone 2-way system to one of the partitions (i.e. one of the two 2-ways), you can get a feel for the LPAR overhead cost. Our test measurement showed a capacity degradation of 3%. That is, two standalone 2-ways have a combined CPW value of 2100. The total CPW values of two 2-ways running on a partitioned four way, as shown above, is 2044. I.e. (2100-2044)/2044 = -3%.

The reasons for the LPAR overhead can be attributed to contention for the shared memory bus on a partitioned system, to the aggregate bandwidth of the standalone systems being greater than the bandwidth of the partitioned system, and to a lower number of system resources configured for a system partition than on a standalone system. For example on a standalone 2-way system the main memory available may be X, and on a partitioned system the amount of main storage available for the 2-way partition is X-2.

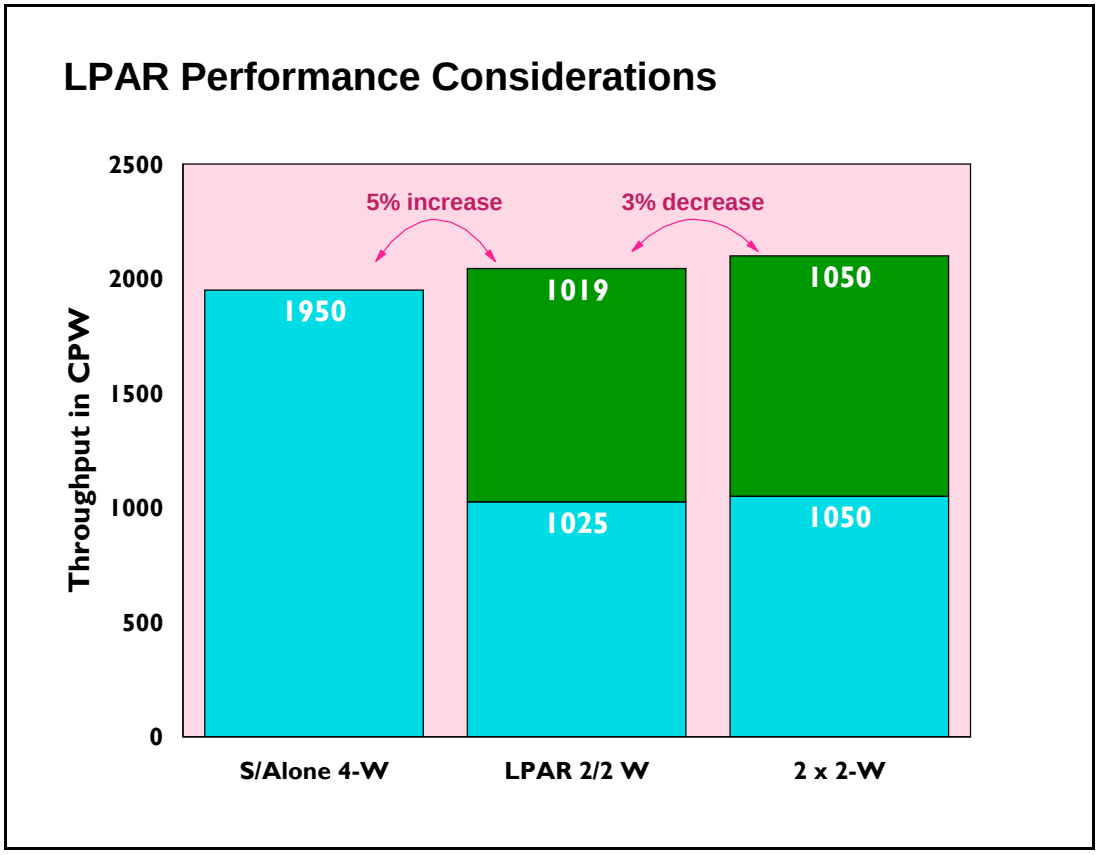



*Figure 18.1. LPAR Performance Measured Against Standalone Systems*

In summary, the measurements on the 4-way system indicate that when a workload can be logically split between two systems, using LPAR to configure two systems will result in system capacities that are greater than when the two applications are run on a single system, and somewhat less than splitting the applications to run on two physically separate systems. The amount of these differences will vary depending on the size of the system and the nature of the application.

## 18.3 Performance on a 12-way system

As the machine size increases we have seen an increase in both the performance of a partitioned system and in the LPAR overhead on the partitioned system. As shown below you will notice that the capacity increase and LPAR overhead is greater on a 12-way system than what was shown above on a 4-way system.

Also note that part of the performance increase of an larger system may have come about because of a reduction in contention within the CPW workload itself. That is, the measurement of the standalone 12-way system required a larger number of users to drive the system's CPU to 70 percent than what is required on a 4-way system. The larger number of users may have increased the CPW workload's internal contention. With a lower number of users required to drive the system's CPU to 70 percent on a standalone 4-way system., there is less opportunity for the workload's internal contention to be a factor in the measurements.

The overall performance of a large system depends greatly on the workload and how well the workload scales to the large system. The overall performance of a large partitioned system is far more complicated because the workload of each partition must be considered as well as how each workload scales to the size of the partition and the resources allocated to the partition in which it is running. While the partitions in a system do not contend for the same main storage, processor, or I/O resources, they all use the same main storage bus to access their data. The total contention on the bus affects the performance of each partition, but the degree of impact to each partition depends on its size and workload.

In order to develop guidelines for partitioned systems, the standard AS/400 Commercial Processing Workload (CPW) was run in several environments to better understand two things. First, how does the sum of the capacity of each partition in a system compare to the capacity of that system running as a single image? This is to show the cost of consolidating systems. Second, how does the capacity of a partition compare to that of an equivalently sized stand-alone system?

The experiments were run on a 12-way 740 model with sufficient main storage and DASD arms so that CPU utilization was the key resource. The following data points were collected:

- Stand-alone CPW runs of a 4-way, 6-way, 8-way, and 12-way
- Total CPW capacity of a system partitioned into an 8-way and a 4-way partition
- Total CPW capacity of a system partitioned into two 6-way partitions
- Total CPW capacity of a system partitioned into three 4-way partitions

The total CPW capacity of a partitioned system is greater than the CPW capacity of the stand-alone 12-way, but the percentage increase is inversely proportional to the size of the largest partition. The CPW workload does not scale linearly with the number of processors. The larger the number of processors, the closer the contention on the main storage bus approached the contention level of the stand-alone 12-way system.

For the partition combinations listed above, the total capacity of the 12-way system increases as shown in the chart below.

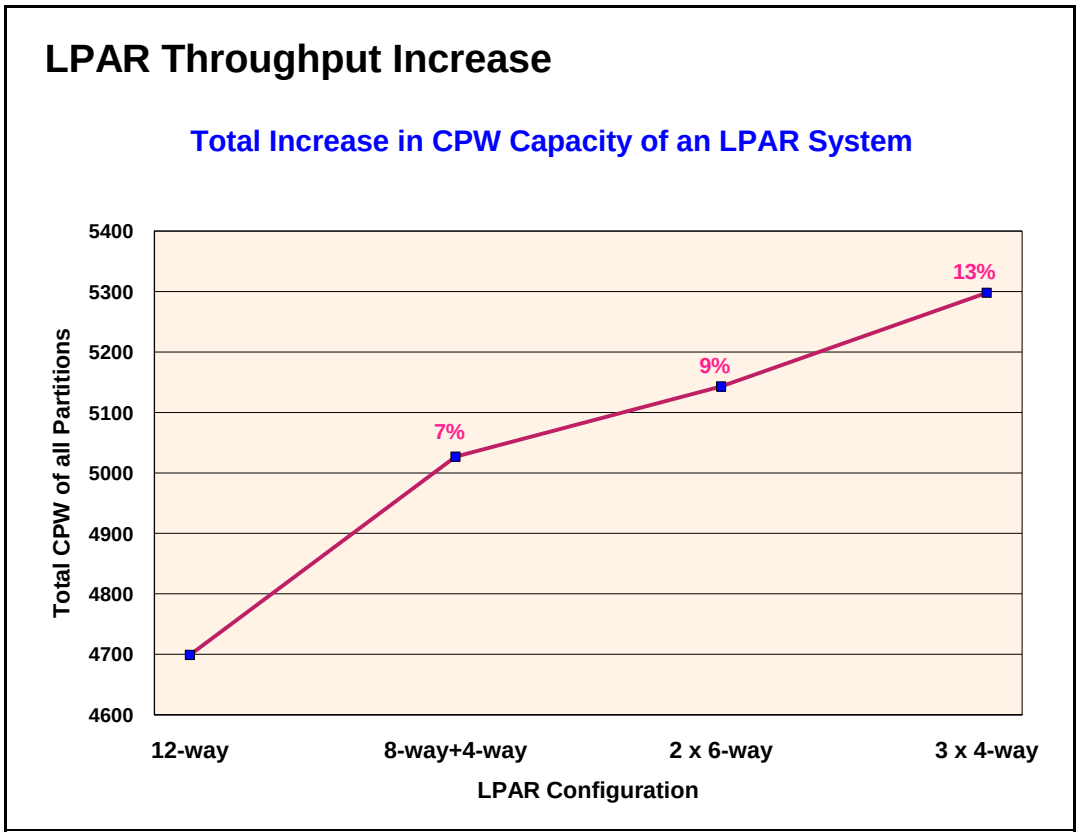



*Figure 18.2. 12 way LPAR Throughput Example*

To illustrate the impact that varying the workload in the partitions has on an LPAR system, the CPW workload was run at an extremely high utilization in the stand-alone 12-way. This high utilization increased the contention on the main storage bus significantly. This same high utilization CPW benchmark was then run concurrently in the three 4-way partitions. In this environment, the total capacity of the partitioned 12-way exceeded that of the stand-alone 12-way by 18% because the total main storage bus contention of the three 4-way partitions is much less than that of a stand-alone 12-way.

The capacity of a partition of a large system was also compared to the capacity of an equally sized stand-alone system. If all the partitions except the partition running the CPW are idle or at low utilization, the capacity of the partition and an equivalent stand-alone system are nearly identical. However, when all of the partitions of the system were running the CPW, then the total contention for the main storage bus has a measurable effect on each of the partitions.

The impact is greater on the smaller partitions than on the larger partitions because the relative increase of the main storage bus contention is more significant in the smaller partitions. For example, the 4-way partition is degraded by 12% when an 8-way partition is also running the CPW, but the 8-way partition is only degraded by 9%. The two 6-way partitions and three 4-way partitions are all degraded by about 8% when they run CPW together. The impact to each partition is directly proportional to the size of the largest partition.

## 18.4 LPAR Measurements

The following chart shows measurements taken on a partitioned 12-way system with the system's CPU utilized at 70 percent capacity. The system was at the V4R4M0 release level.

Note that the standalone 12-way CPW value of 4700 in our measurement is higher than the published V4R3M0 CPW value of 4550. This is because there was a contention point that existed in the CPW workload when the workload was run on large systems. This contention point was relieved in V4R4M0 and this allowed the CPW value to be improved and be more representative of a customer workload when the workload is run on large systems.

*Table 18.1 12-way system measurements*

| LPAR Configuration | Stand alone 12-way CPW | Total LPAR CPW | CPW Increase | LPAR CPW | | | Average LPAR Overhead |
|---|---|---|---|---|---|---|---|
| | | | | Primary | Secondary | Secondary | |
| 8-way, 4-way | 4700 | 5020 | 7% | 3330 | 1690 | n/a | 10 % |
| (2) 6-ways | 4700 | 5140 | 9% | 2605 | 2535 | n/a | 9 % |
| (3) 4-ways | 4700 | 5290 | 13% | 1770 | 1770 | 1750 | 9 % |

While we saw performance improvements on a 12-way system as shown above, part of those improvements may have come about because of a reduction in contention within the CPW workload itself. That is, the measurement of the standalone 12-way system required a larger number of users to drive the system's CPU to 70 percent than what is required on a 4-way system. The larger number of users may have increased the CPW workload's internal contention.

With a lower number of users required to drive the system's CPU to 70 percent on a standalone 4-way system., there is less opportunity for the workload's internal contention to be a factor in the measurements.

The following chart shows our 4-way measurements.

*Table 18.2 4-way system measurements*

| LPAR Configuration | Stand alone 4-way CPW | Total LPAR CPW | CPW Increase | LPAR CPW | | Average LPAR Overhead |
|---|---|---|---|---|---|---|
| | | | | Primary | Secondary | |
| (2) 2-ways | 1950 | 2044 | 5% | 1025 | 1019 | 3 % |

The following chart shows the overhead on n-ways of running a single LPAR partition alone vs. running with other partitions. The differing values for managing partitions is due to the size of the memory nest and the number of processors to manage (n-way size).

*Table 18.3 LPAR overhead per partition*

| Processors | Measured | Projected |
|---|---|---|
| 2 | - | 1.5 % |
| 4 | 3.0 % | - |
| 8 | - | 6.0 % |
| 12 | 9.0 % | - |

The following chart shows projected LPAR capacities for several LPAR configurations. The projections are based on measurements on 1 and 2 way measurements when the system's CPU was utilized at 70 percent capacity. The LPAR overhead was also factored into the projections. The system was at the V4R4M0 release level.

*Table 18.4 Projected LPAR Capacities*

| LPAR Configuration | | Projected LPAR CPW | Projected CPW Increase Over a Standalone 12-way |
|---|---|---|---|
| Number | Processors | | |
| 12 | 1-ways | 5920 | 26 % |
| 6 | 2-ways | 5700 | 21 % |

## 18.5 Summary

On a partitioned system the capacity increases will range from 5% to 26%. The capacity increase will depend on the number of processors partitioned and on the number of partitions. In general the greater the number of partitions the greater the capacity increase.

When consolidating systems, a reasonable and safe guideline is that a partition may have about 10% less capacity than an equivalent stand-alone system if all partitions will be running their peak loads concurrently. This cross-partition contention is significant enough that the system operator of a partitioned system should consider staggering peak workloads (such as batch windows) as much as possible.

# Chapter 19. Miscellaneous Performance Information

## 19.1 Public Benchmarks (TPC-C, SAP, NotesBench, SPECjbb2000, VolanoMark)

iSeries systems have been represented in several public performance benchmarks. The purpose of these benchmarks is to give an indication of relative strength in a general field of computing. Benchmark results can give confidence in a system's capabilities, but should not be viewed as a sole criterion for the purchase or upgrading of a system. We do not include specific benchmark results in this chapter, because the positioning of these results are constantly changing as other vendors submit their own results. Instead, this section will reference several locations on the internet where current information may be found.

A good source of information on many benchmark results can be found at the ideasInternational benchmark page, at http://www.ideasinternational.com/benchmark/bench.html.

**TPC-C Commercial Performance**

The Transaction Processing Performance Council's TPC Benchmark C (TPC-C (**)) is a public benchmark that stresses systems in a full integrity transaction processing environment. It was designed to stress systems in a way that is closely related to general business computing, but the functional emphasis may still vary significantly from an actual customer environment. It is fair to note that the business model for TPC-C was created in 1990, so computing technologies that were developed in subsequent years are not included in the benchmark.

There are two methods used to measure the TPC-C benchmark. One uses multiple small systems connected to a single database server. This implementation is called a "non-cluster" implementation by the TPC. The other implementation method grows this configuration by coupling multiple database servers together in a clustered environment. The benchmark is designed in such a way that these clusters scale far better than might be expected in a real environment. Less than 10% of the transactions touch more than one of the database server systems, and for that small number the cross-system access is typically for only a single record. Because the benchmark allows unrealistic scaling of clustered configurations, we would advise against making comparisons between clustered and non-clustered configurations. All iSeries results and AS/400 results in this benchmark are non-clustered configurations - showing the strengths of our system as a database server.

The most current level of TPC-C benchmark standards is Version 5, which requires the same performance reporting metrics but now requires pricing of configurations to include 24 hr x 7 day a week maintenance rather than 8 hr x 5 day a week and some additional changes in pricing the communication connections. All previous version submissions from reporting vendors have been offered the opportunity to simply republish their results with these new metric ground rules. And as of April, 2001 not all vendors have chosen to republish their results to the new Version 5 standard. iSeries and pSeries has republished.

For additional information on the benchmark and current results, please refer to the TPC's web site at: http://www.tpc.org

**SAP Performance Information**

Several Business Partner companies have defined benchmarks for which their applications can be rated on different hardware and middle ware platforms. Among the first to do this was SAP. SAP has defined a suite of "Standard Application Benchmarks", each of which stresses a different part of SAP's solutions.

The most commonly run of these is the SAP-SD (Sales and Distribution) benchmark. It can be run in a 2-tier environment, where the application and database reside on the same system, or on a 3-tier environment, where there are many application servers feeding into a database server.

Care must be taken to ensure that the same level of software is being run when comparing results of SAP benchmarks. Like most software suppliers, SAP strives to enhance their product with useful functions in each release. This can yield significantly different performance characteristics between releases such as 4.0B, 4.5B, and 4.6C. It should be noted that, although SAP is used as an example here, this situation is not restricted to SAP software.

For more information on SAP benchmarks, go to http://www.sap.com and process a search for Standard Application Benchmarks Published Results.

**NotesBench**

There are several benchmarks that are called "Notesbench xxx". All come from the Notesbench Consortium, a consortium of vendors interested in using benchmarks to help quantify system capabilities using Lotus Domino functions. The most popular benchmark is Notesbench R5 Mail, which is actually a mail and calendar benchmark that was designed around the functions of Lotus Domino Release 5.0. AS/400 and iSeries systems have traditionally demonstrated very strong performance in both capacity and response time in Notesbench results.

For official iSeries audited NotesBench results, see http://www.notesbench.org . (Note: in order to access the NotesBench results you will need to apply for a userid/password through the Notesbench organization. Click on Site Registration at the above address.) An alternate is to refer to the ideasInternational web site listed above.

For more information on iSeries performance in Lotus Domino environments, refer to Chapter 11 of this document.

**SPECjbb2000**

The Standard Performance Evaluation Corporation (SPEC) defined, in June, 2000, a server-side Java benchmark called SPECjbb2000. It is one of the only Java-related benchmarks in the industry that concentrates on activity in the server, rather than a single client. The iSeries architecture is well suited for an object-oriented environment and it provides one of the most efficient and scalable environments for server-side Java workloads. iSeries and AS/400 results are consistently at or near the top rankings for this benchmark.

For more information on SPECjbb2000 and for published results, see http://www.spec.org/osg/jbb2000/

For more information on iSeries performance in Java environments, refer to Chapter 7 of this document.

**VolanoMark**

IBM has chosen the VolanoMark benchmark as another means for demonstrating strength with server-side Java applications. VolanoMark is a 100% Pure Java server benchmark characterized by long-lasting network connections and high thread counts. It is as much a test of tcp/ip strengths as it is of multithreaded, server-side Java strengths. In order to scale well in this benchmark, a solution needs to scale well in tcp/ip, Java-based applications, multithreaded application, and the operating system in general. Additional information on the benchmark can be found at http://www.volano.com/benchmarks.html.

This web site is primarily focused on results for systems that the Volano company measures themselves. These results tend to be for much smaller, Intel-based systems that are not comparable with iSeries servers. The web site also references articles written by other groups regarding their measurements of the benchmark, including AS/400 and iSeries articles. iSeries servers have demonstrated significant strengths in this benchmark, particularly in scaling to large systems.

---

## 19.2 Dynamic Priority Scheduling

On an AS/400 CISC-model, all ready-to-run OS/400 jobs and Licensed Internal Code (LIC) tasks are sequenced on the Task Dispatching Queue (TDQ) based on priority assigned at creation time. In addition, for N-way models, there is a cache affinity field used by Horizontal Licensed Internal Code (HLIC) to keep track of the processor on which the job was most recently active. A job is assigned to the processor for which it has cache affinity, unless that would result in a processor remaining idle or an excessive number of higher-priority jobs being skipped. The priority of jobs varies very little such that the resequencing for execution only affects jobs of the same initially assigned priority. This is referred to as Fixed Priority Scheduling.

For V3R6 and beyond, the new algorithm being used is Dynamic Priority Scheduling. This new scheduler schedules jobs according to "delay costs" dynamically computed based on their time waiting in the TDQ as well as priority. The job priority may be adjusted if it exceeded its resource usage limit. The cache affinity field is no longer used in a N-way multiprocessor machine. Thus, on an N-way multiprocessor machine, a job will have equal affinity for all processors, based only on delay cost.

A new system value, QDYNPTYSCD, has been implemented to select the type of job dispatching. The job scheduler uses this system value to determine the algorithm for scheduling jobs running on the system. The default for this system value is to use Dynamic Priority Scheduling (set to '1'). This scheduling scheme allows the CPU resource to be spread to all jobs in the system.

The benefits of Dynamic Priority Scheduling are:
- No job or set of jobs will monopolize the CPU
- Low priority jobs, like batch, will have a chance to progress
- Jobs which use too much resource will be penalized by having their priority reduced
- Jobs response time/throughput will still behave much like fixed priority scheduling

By providing this type of scheduling, long running, batch-type interactive transactions, such as a query, will not run at priority 20 all the time. In addition, batch jobs will get some CPU resources rather than interactive jobs running at high CPU utilization and delivering response times that may be faster than required.

To use Fixed Priority Scheduling, the system value has to be set to '0'.

**Delay Cost Terminology**

- Delay Cost

  Delay cost refers to how expensive it is to keep a job in the system. The longer a job spends in the system waiting for resources, the larger its delay cost. The higher the delay cost, the higher the priority. Just like the priority value, jobs of higher delay cost will be dispatched ahead of other jobs

of relatively lower delay cost.

- Waiting Time

  The waiting time is used to determine the delay cost of a job at a particular time. The waiting time of a job which affects the cost is the time the job has been waiting on the TDQ for execution.

- Delay Cost Curves

  The end-user interface for setting job priorities has not changed. However, internally the priority of a job is mapped to a set of delay cost curves (see "Priority Mapping to Delay Cost Curves" below). The delay cost curve is used to determine a job's delay cost based on how long it has been waiting on the TDQ. This delay cost is then used to dynamically adjust the job's priority, and as a result, possibly the position of the job in the TDQ.

  On a lightly loaded system, the jobs' cost will basically stay at their initial point. The jobs will not climb the curve. As the workload is increased, the jobs will start to climb their curves, but will have little, if any, effect on dispatching. When the workload gets around 80-90% CPU utilization, some of the jobs on lower slope curves (lower priority), begin to overtake jobs on higher slope curves which have only been on the dispatcher for a short time. This is when the Dynamic Priority Scheduler begins to benefit as it prevents starvation of the lower priority jobs. When the CPU utilization is at a point of saturation, the lower priority jobs are climbing quite a way up the curve and interacting with other curves all the time. This is when the Dynamic Priority Scheduler works the best.

  Note that when a job begins to execute, its cost is constant at the value it had when it began executing. This allows other jobs on the same curve to eventually catch-up and get a slice of the CPU. Once the job has executed, it "slides" down the curve it is on, to the start of the curve.

**Priority Mapping to Delay Cost Curves**

The mapping scheme divides the 99 'user' job priorities into 2 categories:

- User priorities 0-9

  This range of priorities is meant for critical jobs like system jobs. Jobs in this range will NOT be overtaken by user jobs of lower priorities. NOTE: You should generally not assign long-running, resource intensive jobs within this range of priorities.

- User priorities 10-99

  This range of priorities is meant for jobs that will execute in the system with dynamic priorities. In other words, the dispatching priorities of jobs in this range will change depending on waiting time in the TDQ if the QDYNPTYSCD system value is set to '1'.

- The priorities in this range are divided into groups:
  - Priority 10-16
  - Priority 17-22
  - Priority 23-35
  - Priority 36-46

- Priority 47-51
- Priority 52-89
- Priority 90-99

Jobs in the same group will have the same resource (CPU seconds and Disk I/O requests) usage limits. Internally, each group will be associated with one set of delay cost curves. This would give some preferential treatment to jobs of higher user priorities at low system utilization.

With this mapping scheme, and using the default priorities of 20 for interactive jobs and 50 for batch jobs, users will generally see that the relative performance for interactive jobs will be better than that of batch jobs, without CPU starvation.

## Performance Testing Results

Following are the detailed results of two specific measurements to show the effects of the Dynamic Priority Scheduler:

In Table 19.1, the environment consists of the RAMP-C interactive workload running at approximately 70% CPU utilization with 120 workstations and a CPU intensive interactive job running at priority 20.

In Table 19.2 below, the environment consists of the RAMP-C interactive workload running at approximately 70% CPU utilization with 120 workstations and a CPU intensive batch job running at priority 50.

*Table 19.1. Effect of Dynamic Priority Scheduling: Interactive Only*

| | **QDYNPTYSCD = '1' (ON)** | **QDYNPTYSCD = '0'** |
|---|---|---|
| Total CPU Utilization | 93.9% | 97.8% |
| Interactive CPU Utilization | 77.6% | 82.2% |
| RAMP-C Transactions per Hour | 60845 | 56951 |
| RAMP-C Average Response Time | 0.32 | 0.75 |
| Priority 20 CPU Intensive Job CPU | 21.9% | 28.9% |

*Table 19.2. Effect of Dynamic Priority Scheduling: Interactive and Batch*

| | **QDYNPTYSCD = '1' (ON)** | **QDYNPTYSCD = '0'** |
|---|---|---|
| Total CPU Utilization | 89.7% | 90.0% |
| Interactive CPU Utilization | 56.3% | 57.2% |
| RAMP-C Transactions per Hour | 61083 | 61692 |
| RAMP-C Average Response Time | 0.30 | 0.21 |
| Batch Priority 50 Job CPU | 15.0% | 14.5% |
| Batch Priority 50 Job Run Time | 01:06:52 | 01:07:40 |

## Conclusions/Recommendations

- When you have many jobs running on the system and want to ensure that no one CPU intensive job 'takes over' (see Table 19.1 above), Dynamic Priority Scheduling will give you the desired result. In this case, the RAMP-C jobs have higher transaction rates and faster response times, and the priority 20 CPU intensive job consumes less CPU.

- Dynamic Priority Scheduling will ensure your batch jobs get some of the CPU resources without significantly impacting your interactive jobs (see Table 96). In this case, the RAMP-C workload gets

less CPU utilization resulting in slightly lower transaction rates and slightly longer response times. However, the batch job gets more CPU utilization and consequently shorter run time.

- It is recommended that you run with Dynamic Priority Scheduling for optimum distribution of resources and overall system performance.

For additional information, refer to the *Work Management Guide*.

## 19.3 Main Storage Sizing Guidelines

To take full advantage of the performance of the new AS/400 Advanced Series using PowerPC technology, larger amounts of main storage are required. To account for this, the new models are provided with substantially more main storage included in their base configurations. In addition, since more memory is required when moving to RISC, memory prices have been reduced.

The increase in main storage requirements is basically due to two reasons:

- When moving to the PowerPC RISC architecture, the number of instructions to execute the same program as on CISC has increased. This does not mean the function takes longer to execute, but it does result in the function requiring more main storage. This obviously has more of an impact on smaller systems where fewer users are sharing the program.

- The main storage page size has increased from 512 bytes to 4096 bytes (4KB). The 4KB page size is needed to improve the efficiency of main storage management algorithms as main storage sizes increase dramatically. For example, 4GB of main storage will be available on AS/400 Advanced System model 530.

  The impact of the 4KB page size on main storage utilization varies by workload. The impact of the 4KB page size is dependent on the way data is processed. If data is being processed sequentially, the 4KB page size will have little impact on main storage utilization. However, if you are processing data randomly, the 4KB page size will most likely increase the main storage utilization.

## 19.4 Memory Tuning Using the QPFRADJ System Value

The Performance Adjustment support (QPFRADJ system value) is used for initially sizing memory pools and managing them dynamically at run time. In addition, the CHGSHRPOOL and WRKSHRPOOL commands allow you to tailor memory tuning parameters used by QPFRADJ. You can specify your own faulting guidelines, storage pool priorities, and minimum/maximum size guidelines for each shared memory pool. This allows you the flexibility to set unique QPFRADJ parameters at the pool level.

For a detailed discussion of what changes are made by QPFRADJ, see the Work Management Guide. What follows is a description of some of the affects of this system value and some discussion of when the various settings might be appropriate.

**When the system value is set to 1**, adjustments are made to try to balance the machine pool, base pool, spooling pool, and interactive pool at IPL time. The machine pool is based on the amount of storage needed for the physical configuration of the system; the spool pool is fairly small and reflects the number

of printers in the configuration. 70% of the remaining memory is allocated to the interactive pool; 30% to the base pool.

A QPFRADJ value of 1 ensures that memory is allocated on the system in a way that the system will perform adequately at IPL time. It does not allow for reaction to changes in workload over time. In general, this value is avoided unless a routine will be run shortly after an IPL that will make adjustments to the memory pools based on the workload.

**When the system value is set to 2**, adjustments are made as described, plus dynamic changes are made as changes in workload occur. In addition to the pools mentioned above, shared pools (*SHRPOOLxxx) are also managed dynamically. Adjustments are based on the number of jobs active in the subsystem using the pool, the faulting rates in the pool, and on changes in the workload over the course of time.

This is a good option for most environments. It attempts to balance system memory resources based on the workload that is being run at the time. When workload changes occur, such as time-of-day changes when one workload may increase while another may decrease, memory resources are gradually shifted to accommodate the heaviest loads.

**When the system value is set to 3**, adjustments are only made during the runtime, not as a result of an IPL.

This is a good option if you believe that your memory configuration was reasonable prior to scheduling an IPL. Overall, having the system value set to 2 or 3 will yield a similar effect for most environments.

**When the system value is set to 0,** no adjustments are made. This is a good option if you plan on managing the memory by yourself. Examples of this may be if you know times when abrupt changes in memory are likely to be required (such as a difference between daytime operations and nighttime operations) or when you want to always have memory available for specific, potentially sporadic work, even at the expense of not having that memory available for other work. It should be noted, however, that this latter case can also be covered by using a private memory pool for this work. The QPFRADJ system value only affects tuning of system-supplied shared pools.

---

## 19.5 Additional Memory Tuning Techniques

### Expert Cache

Normally, the system will treat all data that is brought into a memory pool in a uniform way. In a purely random environment, this may be the best option. However, there are often situations where some files are accessed more often than others or when some are accessed in blocks of information instead of randomly. In these situations, the use of "Expert Cache" may improve the efficiency of the memory in a pool. Expert Cache is enabled by changing the pool attribute from *FIXED to *CALC. One advantage for using Expert Cache (*CALC) is that the system dynamically determines which objects should have larger blocks of data brought into main storage. This is based on how frequently the object is accessed. If the object is no longer accessed heavily, the system automatically makes the storage available for other objects that are accessed. If the newly accessed objects then become heavily accessed, the objects have larger blocks of data placed in main storage.

Expert Cache is often the best solution for batch processing, when relatively few files may be accessed in large blocks at a time or in sequential order. It is also beneficial in many interactive environments when

files of differing characteristics are being accessed. The pool attribute can be changed from *FIXED to *CALC and back at any time, so making a change and evaluating its affect over a period of time is a fairly safe experiment.

More information about Expert Cache can be found in the Work Management guide.

In some situations, you may find that you can achieve better memory utilization by defining the caching characteristics yourself, rather than relying on the system algorithms. This can be done using the QWCCHGTN (Change Pool Tuning Information) API, which is described in the Work Management API reference manual. This API was provided prior to the offering of the *CALC option for the system. It is still available for use, although most situations will see relatively little improvement over the *CALC option and it is quite possible to achieve less improvement than with *CALC. When the API is used to adjust the pool attribute, the value that is shown for the pool is USRDFN (user defined).

**SETOBJACC (Set Object Access)**

In some cases, the object access performance is improved when the user manually defines (names a specific object) which object is placed into main storage. This can be achieved with the SETOBJACC command. This command will clear any pages of an object that are in other storage pools and moves the object to the specified pool. If the object is larger than the pool, the first portions of the object are replaced with the later pages that are moved into the pool. The command reports on the current amount of storage that is used in the pool.

If SETOBJACC is used when the QPFRADJ system value is set to either 2 or 3, the pool that is used to hold the object should be a private pool so that the dynamic adjustment algorithms do not shrink the pool because of the lack of job activity in the pool.

**Large Memory Systems**

Normally, you will use memory pools to separate specific sets of work, leaving all jobs which do a similar activity in the same memory pool. With today's ability to configure many gigabytes of mainstore, you may also find that work can be done more efficiently if you divide large groups of similar jobs into separate memory pools. This may allow for more efficient operation of the algorithms which need to search the pool for the best candidates to purge when new data is being brought in. Laboratory experiments using the I/O intensive CPW workload on a fully configured 24-way system have shown about a 2% improvement in CPU utilization when the transaction jobs were split among pools of about 16GB each, rather than all running in a single memory pool.

## 19.6 User Pool Faulting Guidelines

Due to the large range of AS/400 processors and due to an ever increasing variance in the complexity of user applications, paging guidelines for user pools are no longer published. Even the system wide guidelines are just that...guidelines. Each customer needs to track response time, throughput, and cpu utilization against the paging rates to determine a reasonable paging rate.

There are two choices for tuning user pools:

1. Set system value QPFRADJ = 2 or 3, as described earlier in this chapter.
2. Manual tuning. Move storage around until the response times and throughputs are acceptable. The rest of this section deals with how to determine these acceptable levels.

To determine a reasonable level of page faulting in user pools, determine how much the paging is affecting the interactive response time or batch throughput. These calculations will show the percentage of time spent doing page faults.

The following steps can be used: (all data can be gathered w/STRPFRMON and printed w/PRTSYSRPT). The following assumes interactive jobs are running in their own pool, and batch jobs are running in their own pool.

**Interactive**:

1. flts = sum of database and non-database faults per second during a meaningful sample interval for the interactive pool.

2. rt = interactive response time for that interval.

3. diskRt = average disk response time for that interval.

4. tp = interactive throughput for that interval in transactions per second. (transactions per hour/3600 seconds per hour)

5. fltRtTran = diskRt * flts / tp = average page faulting time per transaction.

6. flt% = fltRtTran / rt * 100 = percentage of response time due to

7. If flt% is less than 10% of the total response time, then there's not much potential benefit of adding storage to this interactive pool. But if flt% is 25% or more of the total response time, then adding storage to the interactive pool may be beneficial (see NOTE below).

**Batch**:

1. flts = sum of database and non-database faults per second during a meaningful sample interval for the batch pool.

2. flt% = flts * diskRt X 100 = percentage of time spent page faulting in the batch pool. If multiple batch jobs are running concurrently, you will need to divide flt% by the number of concurrently running batch jobs.

3. batchcpu% = batch cpu utilization for the sample interval. If higher priority jobs (other than the batch jobs in the pool you are analyzing) are consuming a high percentage of the processor time, then flt% will always be low. This means adding storage won't help much, but only because most of the batch time is spent waiting for the processor. To eliminate this factor, divide flt% by the sum of flt% and batchcpu%. That is: **newflt% = flt% / (flt% + batchcpu%)**
   This is the percentage of time the job is spent page faulting compared to the time it spends at the processor.

4. Again, the potential gain of adding storage to the pool needs to be evaluated. If flt% is less than 10%, then the potential gain is low. If flt% is greater than 25% then the potential gain is high enough to warrant moving main storage into this batch pool.

**NOTE**:

It is very difficult to predict the improvement of adding storage to a pool, even if the potential gain calculated above is high. There may be instances where adding storage may not improve anything because of the application design. For these circumstances, changes to the application design may be necessary.

Also, these calculations are of limited value for pools that have expert cache turned on. Expert cache can reduce I/Os given more main storage, but those I/Os may or may not be page faults.

---

## 19.7 AS/400 NetFinity Capacity Planning

Performance information for AS/400 NetFinity attached to a V4R1 AS/400 is included below. The following NetFinity functions are included:

- Time to collect software inventory from client PCs
- Time to collect hardware inventory from client PCs

The figures below illustrate the time it takes to collect software and hardware inventory from various numbers of client PCs. This test was conducted using the Rochester development site, during normal working hours with normal activity (i.e.. not a dedicated environment). This environment consists of:

- 16 and 4Mb token ring LANs (mostly 16)
- LANs connected via routers and gateways
- Dedicated AS/400
- TCP/IP
- Client PCs varied from 386s to Pentiums (mostly 100 MHz with 32MB memory), using OS/2, Windows/95 and NT
- About 20K of data was collected, hardware and software, for each client

While these tests were conducted in a typical work environment, results from other environments may vary significantly from what is provided here.

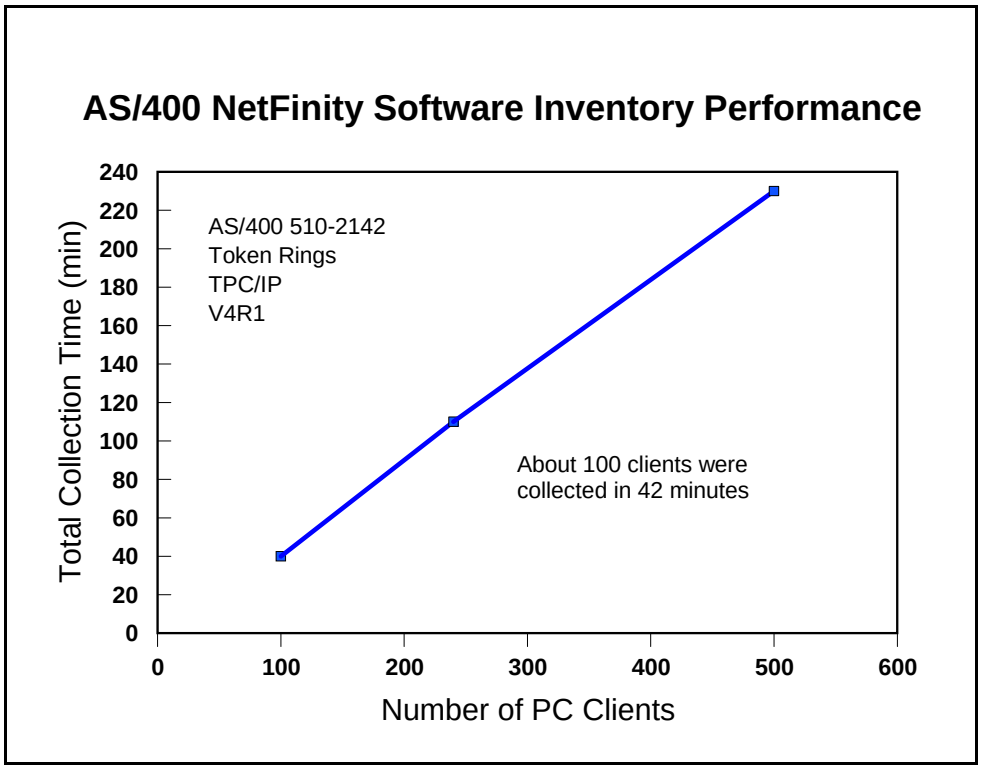



Figure 19.1. AS/400 NetFinity Software Inventory Performance

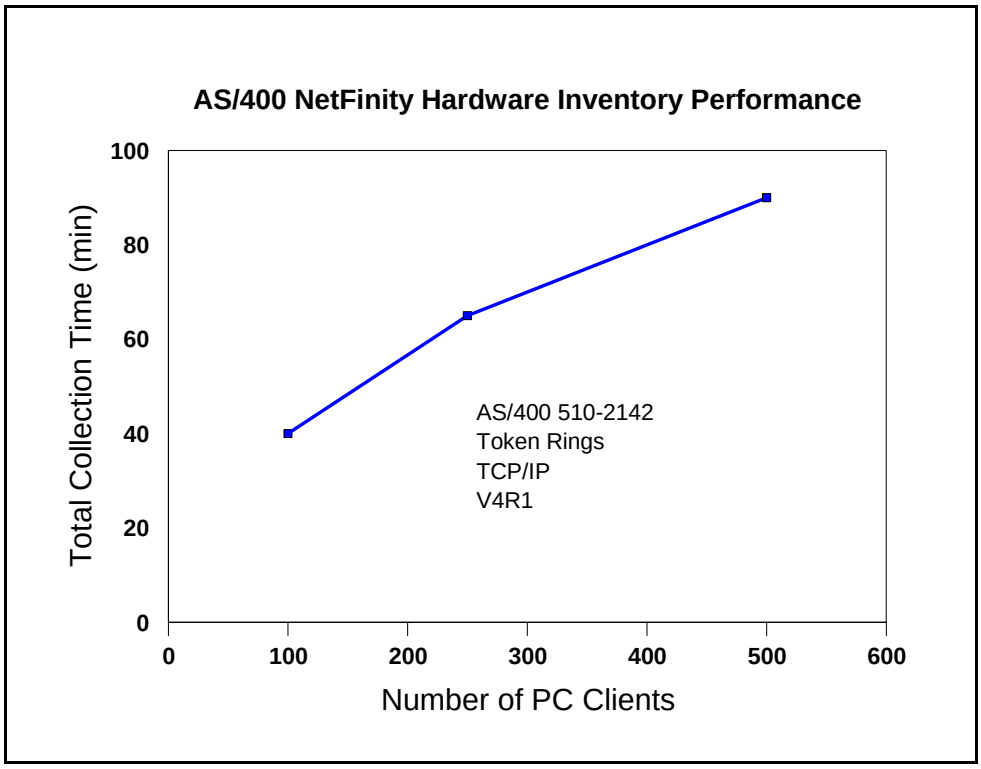



Figure 19.2. AS/400 NetFinity Hardware Inventory Performance

**Conclusions/Recommendations for NetFinity**

1. The time to collect hardware or software information for a number of clients is fairly linear.

2. The size of the AS/400 CPU is not a limitation. Data collection is performed at a batch priority. CPU utilization can spike quite high (ex. 80%) when data is arriving, but in general is quite low (ex. 10%).

3. The LAN type (4 or 16Mb Token Ring or Ethernet) is not a limitation. Hardware collection tends to be more chatty on the LAN than software collection, depending on the hardware features.

4. The communications protocol (IPX, TCP/IP, or SNA) is not a limitation.

5. Collected data is automatically stored in a standard DB2/400 database file, accessible by SQL and other APIs.

6. Collection time depends on clients being powered-on and the needed software turned on. The server will retry 5 times.

7. The number of jobs on the server increases during collection and decreases when not needed.

# Chapter 20. General Performance Tips and Techniques

This section's intent is to cover a variety of useful topics that "don't fit" in the document as a whole, but provide useful things that customers might do or deal with special problems customers might run into on iSeries. It may also contain some general guidelines.

## 20.1 Adjusting Your Performance Tuning for Threads

### History

Historically, the iSeries and AS/400 programmers have not had to worry very much about threads. True, they were introduced into the machine some time ago, but the average RPG application does not use them and perhaps never will, even if it is now allowed. Multiple-thread jobs have been fairly rare. That means that those who set up and organize AS/400 subsystems (e.g. QBATCH, QINTER, MYOWNSUBSYSTEM, etc.) have not had to think much about the distinction between a "job" and a "thread."

### The Coming Change

But, threads are a good thing and so applications are increasingly using them. Especially for customers deploying (say) a significant new Java application, or Domino, a machine with the typical one-thread-per-job model may suddenly have dozens or even hundreds of threads in a particular job. Unfortunately, they are distinct ideas and certain AS/400 commands carefully distinguish them. If iSeries System Administrators are careless about these distinctions, as it is so easy to do today, poor performance can result as the system moves on to new applications such as Lotus Domino or especially Java.

With Java generally, and with certain applications, it will be commonplace to have multiple threads in a job. That means taking a closer look at some old friends: MAXACT and MAXJOB.

Recall that every subsystem has at least one pool entry. Recall further that, in the subsystem description itself, the pool number is an arbitrary number. What is more important is that the arbitrary number maps to a particular, real storage pool (*BASE, *SHRPOOL1, etc.). When a subsystem is actually started, the actual storage pool (*SHRPOOL1), if someone else isn't already using it, comes to life and obtains its storage.

However, storage pools are about more than storage. They are also about job and thread control. Each pool has an associated value called MAXACT that also comes into play. No matter how many subsystems share the pool, MAXACT limits the total number of threads able to reside and execute in the pool. Note that this is *threads* and not *jobs*.

Each subsystem, also, has a MAXJOBS value associated with it. If you reach that value, you are not supposed to be able to start any more jobs in the subsystem. Note that this is a *jobs* value and not a *threads* value. Further, within the subsystem, there are usually one or more JOBQs in the subsystem. Within each entry you can also control the number of jobs using a parameter. Due to an unfortunate turn in history, this parameter, which might more logically be called MAXJOBS today is called MAXACT. However, it controls *jobs*, not *threads*.

**Problem**

It is too easy to use the overall pool's value of MAXACT as a surrogate for controlling the number of Jobs. That is, you can forget the distinction between jobs and threads and use MAXACT to control the activity in a storage pool. But, you are not controlling jobs; you are controlling threads.

It is also too easy to have your existing MAXACT set too low if your existing QBATCH subsystem suddenly sees lots of new Java threads from new Java applications.

If you make this mistake (and it is easy to do), you'll see several possible symptoms:

- Mysterious failures in Java. If you set the value of MAXACT really low, certainly as low as one, sometimes Java won't run, but it also won't always give a graceful message explaining why.
- Mysterious "hangs" and slowdowns in the system. If you don't set the value pathologically low, but still too low, the system will function. But it will also dutifully "kick out" threads to a limbo known as "ineligible" because that's what MAXACT tells it to do. When MAXACT is too low, the result is useless wait states and a lot of system churn. In severe cases, it may be impossible to "load up" a CPU to a high utilization and/or response times will substantially increase.
- Note carefully that this can happen as a result of an upgrade. If you have just purchased a new machine and it runs slower instead of faster, it may be because you're using "yesterday's" limits for MAXACT

If you're having threads thrown into "ineligible", this will be visible via the WRKSYSSTS command. Simply bring it up, perhaps press PF11 a few times, and see if the Act->Inel is something other than zero. Note that other transitions, especially Act->Wait, are normal.

**Solution**

Make sure the *storage pool's* MAXACT is set high enough for each individual storage pool. A MAXACT of *NOMAX will sometimes work quite well, especially if you use MAXJOBS to control the amount of working coming into each subsystem.

Use CHGSHRPOOL to change the number of *threads* that can be active in the pool (note that multiple subsystems can share a pool):

CHGSHRPOOL ACTLVL(newmax)

Use MAXJOB in the subsystem to control the amount of outstanding work in terms of *jobs*:

CHGSBSD QBATCH MAXJOBS(newmax)

Use the Job Queue Entry in the subsystem to have even finer control of the number of jobs:

CHGJOBQE SBSD(QBATCH) JOBQ(QBATCH) MAXACT(newqueue job maximum)

Note in this particular case that MAXACT does refer to jobs and not threads.

## 20.2 General Performance Guidelines -- Effects of Compilation

In general, the higher the optimization, the less easy the code will be to debug. It may also be the case that the program will do things that are initially confusing.

### In-lining

For instance, suppose that ILE Module A calls ILE Module B. ILE Module B is a C program that does allocation (malloc/free in C terms). However, in the right circumstances, compiler optimization will "inline" Module B. In-lining means that the code for B is not called, but it is copied into the calling module instead and then further optimized. So, for at least Module A, then, the "in-lined" Module B will cease to be an individual compiled unit and simply have its code copied, verbatim, into A.

Accordingly, when performance traces are run, the allocation activity of Module B will show up under Module A in the reports. Exceptions would also report the exception taking place in Module A of Program X.

In-lining of "final" methods is possible in Java as well, with similar implications.

### Optimization Levels

Most of the compilers and Java support a reasonably compatible view of optimization. That is, if you specify OPTIMIZE(10) in one language, it performs similar levels of optimization in another language, including Java's CRTJVAPGM command. However, these things can differ at the detailed level. Consult the manuals in case of uncertainty.

Generally:

- OPTIMIZE(10) is the lowest and most debuggable.
- OPTIMIZE(20) is a trade-off between rapid compilation and some minimal optimization
- OPTIMIZE(30) provides a higher level of optimization, though it usually avoids the more aggressive options. This level can debug with difficulty.
- OPTIMIZE(40) provides the highest level of optimization. This includes sophisticated analysis, "code motion" (so that the execution results are what you asked for, but not on a statement-by-statement basis), and other optimizations that make debugging difficult. At this level of optimization, the programmer must pay stricter attention to the manuals. While it is surprisingly often irrelevant in actual cases, many languages have specific definitions that allow latitude to highly optimized compilers to do or, more importantly, "not do" certain functions. If the coder is not aware of this, the code may behave differently than expected at high optimization levels.

### LICOPT

A new option has been added to most ILE Languages called LICOPT. This allows language specific optimizations to be turned on and off as individual items. A full description of this is well beyond the scope of this paper, but those interested in the highest level of performance and yet minimizing potential difficulties with specific optimization types would do well to study these options.

## 20.3 How to Design for Minimum Main Storage Use (especially with Java, C, C++)

The iSeries family has added popular languages whose usage continues to increase -- Java, C, C++. These languages frequently use a different kind of storage -- heap storage.

Many iSeries programmers, with a background in RPG or COBOL are unaware of the influence this may have on storage consumption. Why? Simply because these languages, by their nature, do not make much if any use of the heap. Meanwhile, C, C++, and Java very typically do.

The implications can be very profound. Many programmers are unclear about the tradeoffs and, when reducing memory usage, frequently attack the wrong problem. It is surprisingly easy, with these languages, to spend many megabytes and even hundreds of megabytes of main storage without really understanding how and why this was done.

Conversely, with the right understanding of heap storage, a programmer might be able to solve a much larger problem on the identical machine.

### Theory -- and Practice

This is one place where theory really matters. Often, programmers wonder whether a theory applies in practice. After surveying a set of applications, we have concluded that the theory of memory usage applies very widely in practice.

In computer science theory, programmers are taught to think about how many “entities” there are, not how big the entity is. It turns out that controlling the number of entities matters most in terms of controlling main storage -- and even processor usage (it costs some CPU, after all, to *have* and *initialize* storage in the first place). This is largely a function of design, but also of storage layout. It is also knowing which storage is critical and which is not. Formally, the literature talks about:

Order(1) -- about one entity per system
Order(N) -- about “N” entities, where “N” are things like number of data base records, Java objects, and like items.
Order(N log N) -- this can arise because there is a data base and it has an accompanying index.
Order(N squared) -- data base joins of two data bases can produce this level of storage cost

Note the emphasis on “about.” It is the number of entities in relation to the elements of the problem that count. An element of the problem is not a program or a subsystem description. Those are Order(1) costs. It is a data base record, objects allocated from the heap inside of loops, or anything like these examples. In practice, Order(N) storage predominates, so this paper will concentrate on Order(N).

Of course, one must eventually get down to actual sizes. Thus, one ends up with actual costs that get Order(N) estimated like this:

ActualCostForOrder(1) = a

ActualCostInBytes(N) = a + (b x N)

Where a and b are constants. “a” is determined by adding up things like the static storage taken up by the application program. “b” is the size of the data base record plus the size of anything else, such as a Java object, that is created one entity per data base record. In some applications, “N” will refer to some freestanding fact, like the maximum number of concurrent web serving operations or the number of outstanding new orders being processed.

However, the number of data base records will very often be the source of “N.” Of course, with multiple data base files, there may be more than one actual “N”. Still, it is usually true that the record count of one file compared to another will often be available as a ratio. For instance, one could have an “Order” record and average of three and a half “Order Detail” records. As long as the ratio is reasonably stable or can be planned at a stable value, it is a matter of convention which is picked to be “N” in that case; one merely adjusts “b” in the above equation to account for what is picked for “N”.

### System Level Considerations

In terms of the computer science textbooks, we are largely done. But, for someone in charge of commercial application deployment, there is one more practical thing to consider: Jobs and those newer items that now often come with them, threads.

Formally, if there is only one job or thread, then these are part of the Order(1) storage. If there are many, they end up proportional to N (e.g. One job for every 100,000 active records) and so are part of the Order(N) storage cost.

However, it is frequently possible to adjust these based on observed performance effects; the ratio to N is not entirely fixed. So, it is remains of interest to segregate these when planning storage. So, while they will not appear on the formal computer science literature, this paper will talk about Order(j) and Order(t) storage.

### Typical Storage Costs

Here are typical things in modern systems and where they ordinarily sit in terms of their “entity” relationships.

| Order(1) | Order(j) | Order(t) | Order(N) |
|---|---|---|---|
| ILE and OS/400 Programs | Just In Time compiled programs (Java *JIT) | Java threads | Data Base Records and IFS file records |
| Subsystem Descriptions | Total Job Storage | File Buffers of all kinds | Java (and C/C++) objects |
| Direct Execution Java Programs | Static storage from RPG and COBOL. Static final in Java. | SQL Result Set (nonrecord) | Operating System copies (e.g. Data Base) copies of application records |
| System values | Java Virtual Machine and most WebSphere storage | Program stack storage | SQL records in a result set |

**A Brief Example**

To show these concepts, consider a simple example.

Part of a financial system has three logical elements to deal with:

1. An order record (order summary including customer information, sales tax, etc.)
2. An order detail record (individual purchased items, quantities, prices).
3. A table containing international currency rates of exchange between two arbitrary countries.

Question: What is more important? Reducing the cost of the detail record by a couple of bytes, or reducing the currency table from a cost of N squared (where "N" is the number of countries) to 2 times N.

There are two obvious implementations of the currency table:

1. Implement the table as a two dimensional array such that CurrencyExchangei,j will give the exchange between $country_i$ and $country_j$ for all countries.

2. Implement the table as a single dimension array with the *i*th element being the exchange rate between $country_i$ and the US dollar. One can convert to any country simply by converting twice; once to dollars and once to the other currency.

Clearly, the second is more storage efficient.

Now consider the first problem. The detail record looks like this:

Quantity as a four byte number (9B or 10B in RPG terms).
Name of the item (up to 60 characters)
Price of the item (as a zoned decimal field, 15 total digits with two decimal points).

A simple scrub would give:

Quantity as a two byte number (4B in RPG terms).
Name of the item (probably still 60 characters)
Price of the item (as a packed decimal field, probably 10 total digits with two decimal points).

How practical this change would be, if it represented a large, existing data base, would be a separate question. If this is at the initial design, however, this is an easy change to make.

Boundary considerations. In Java, we are done because Java will order the three entities such that the least amount of space is wasted. In C and C++, it might be possible to lay out the storage entities such that the compiler will not introduce padding between elements. In this particular example, the order given above would work out well.

**Which is more important?**

Reading the above superficially, one would expect the currency table improvement to matter most. There was a reduction from an N squared to an 2 times N relationship. However, this cannot be right. In fact, the number of countries is not "N" for this problem. "N" is the number of outstanding orders, a number that is likely in a practical system to be much larger than the number of countries. More critically, the number of countries is essentially fixed. Yes, the number of countries in the world change from time to time. But, of course, this is not the same degree of change as order records in an order entry system. In fact, the currency table is part of the Order(1) storage. The choice between 2 times N and N squared should be based on whatever is operationally simpler.

Perform this test to know what "N" really is: If your department merged with a department of the same size, doing the same job, which storage requirements would double? It is these factors that reveal what the value of "N" is for your circumstances.

And, of course, the detail order record would be one such item. So, where are the savings? The above recommendations will save 9 bytes per record. If you write the code in RPG, this does not seem like much. That would be 9 bytes times the number of jobs used to process the incoming records. After all, there is only one copy of the record in a typical RPG program.

However, one must account for data base. Especially when accessing the records through an index of some kind, the number of records data base will keep laying about will be proportional to "N" -- the total number of outstanding orders. In Java, this can be even more clear-cut. In some Java programs, one processes records one at a time, just as in RPG. The most straightforward case is some sort of "search" for a particular record. In Java, this would look roughly the same as RPG and potentially consume the same storage.

However, Java can also use the power of the heap storage to build huge networks of records. A custom sort of some kind is one easy example of this.

In that case, it is easy for Java to contain the summary record and "dozens" of detail records, all at once, all connected together in a whole variety of ways. If necessary, modern applications might bring in the entire file for the custom sort function, which would then have a peak size at least as large as the data base file(s) itself or themselves.

Once you get above a couple hundred records, even in but one application, the storage savings for the record scrub will swamp the currency table savings. And, since one might have to buy for peak storage usage, even one application that references thousands of detail records would be enough to tip the scale.

**A Short but Important Tip about Data Base**

One thing easily misunderstood is variable length characters. At first, one would think every character field should be variable length, especially if one codes in Java, where variable length data is the norm.

However, when one considers the internals of data base, a field ought to be ten to twenty bytes long before variable length is even considered. The reason is, there is a cost of about ten bytes per record for the first variable length field. Obviously, this should not be introduced to "save" a few bytes of data.

Likewise, the "ALLOCATE" value should be understood (in OS/400 SQL, "ALLOCATE" represents the minimum amount of a variable record always present). Getting this right can improve performance. Getting it wrong simply wastes space. If in doubt, do not specify it at all.

### A Final Thought About Memory and Competitiveness

The currency storage reduction example remains a good one -- just at the wrong level of granularity. Avoiding a SQL join that produces N squared records would be an example where the 2 times N alternative, if available, saves great amounts of storage.

But, more critically, deploying the least amount of Order(N) storage in actual implementation is a competitive advantage for your enterprise, large or small. Reducing the size of each N in main storage (or even on disk) eventually means more "things" in the same unit of storage. That is more competitive whether the cost of main storage falls by half tomorrow or not. More "things" per byte is always an advantage. It will always be cheaper. Your competitor, after all, will have access to the same costs. The question becomes: Who uses it better?

## 20.4 Hardware Multi-threading (HMT)

Hardware multi-threading is a facility present in several iSeries processors. The eServer i5 models instead have the Simultaneous Multi-threading (SMT) facility, which are discussed in the SMT white paper at the following website: http://www-1.ibm.com/servers/eserver/iseries/perfmgmt/pdf/SMT.pdf.

HMT is mentioned here primarily to compare-and-contrast with the SMT. Moreover, several system facilities operate slightly differently on HMT machines versus SMT machines and these differences need some highlighting.

### HMT Described

Broadly, HMT exploited the concept that modern processors are often quite fast relative to certain memory accesses.

Without HMT, a modern CPU might spend a lot of time stalled on things like cache misses. In modern machines, the memory can be a considerable distance from the CPU, which translates to more cycles per fetch when a cache miss occurs. The CPU idles during such accesses.

Since many OS/400 applications feature database activity, cache misses often figured noticeably in the execution profile. Could we keep the CPU busy with something else during these misses?

HMT created two securely segregated streams of execution on one physical CPU, both controlled by hardware. It was created by replicating key registers including another instruction counter. Generally, there is a distinction between the one physical processor and its two logical processors. However, for HMT, the customer seldom sees any of this as the various performance facilities of the system continue to report on a physical CPU basis.

Unlike SMT, HMT allows only one instruction stream to execute at a time. But, if one instruction stream took a cache miss, the hardware switches to the other instruction stream (hence, "hardware multi-threading" or, some say, "hardware multi-tasking"). There would, of course, be times when both were waiting on cache misses, or, conversely, applications that hardly ever had misses. Yet, on the whole, the facility works well for OS/400 applications.

The system value QPRCMLTTSK was introduced in order to turn HMT on or off. This could only take affect when the whole system was IPLed, so (for clarity) one should change the system value itself shortly before a full system IPL. The default is to have it set on ('1').

Generally, in most commercial workloads, HMT enabled ('1') gives gains in throughput between 10 and 25 percent, often without impact to response time.

In rare cases, HMT results in losses rather than gains.

### HMT and SMT Compared and Contrasted

Some key similarities and differences are:

| HMT Feature | SMT Feature |
|---|---|
| •HMT is can be turned on and off only by a whole system IPL. | •SMT can be turned on and off dynamically at any time. No IPL required |
| •All partitions have the same value for HMT | •SMT, because it is more dynamic, the SMT state need not be identical across partitions. |
| •HMT executes only one instruction stream at a time. | •SMT allows multiple streams of execution simultaneously. |
| •CPU utilization measurements are not greatly affected by HMT. | •SMT complicates the question of measuring CPU utilization. |
| •System performance counters and CPU utilization values continue to be reported on a physical CPU basis. | •SMT machines continue to report data on a physical processor basis, but some of the measurements are harder to interpret (reporting on a logical CPU basis would be no better). |
| •HMT operation is controlled by the system value QPRCMLTTSK ("1" means active, "0" means inactive) | •SMT has three values for QPRCMLTTSK ("0" for off, also called "ST mode", "1" for on, and "2" for "controlled" where OS/400 decides, dynamically, whether to be in ST or SMT mode. |
| •HMT needs a full IPL for the change to QPRCMLTTSK to be activated. | •SMT can allow QPRCMLTTSK to change at any time. |
| •HMT typically improves throughput by 10 to 25 per cent. | •SMT can improve throughput up to 40 per cent, in rare cases, higher. |
| | |

### Models With/Without HMT

Not all prior models have HMT. In fact, some recent models have neither HMT nor SMT.

The following models have HMT available:

- 270, 800, 810, 820, 830, 840

The following have neither SMT nor HMT:

- 825, 870, 890

Earlier models than the 270 or 820 series (e.g. 170, 7xx, etc.) did not have either HMT nor SMT.

## 20.5 POWER6 520 Memory Considerations

Because of the design of the Power6 520 system, there are some key factors with the memory subsystem that one should keep in mind when sizing this system. The Power6 520, unlike the Power6 570, has no L3 cache, which does have an effect on memory sensitive workloads, like Java applications for instance. Having no L3 cache makes memory speed, or the bandwidth rating in megabytes per second, even more critical for memory sensitive workloads. The Power6 520 has 8 memory DIMM slots, which are positioned in groups of four behind each of the Power6-SCM modules and each group of four will be referred to as a quad for this discussion. The available number of active memory slots depends on the Processor Feature Code of the system.

When only one SCM module is installed, only one quad of memory is active and all slots must contain DIMMs of the same size and speed. When two SCM modules are installed (except in the case of the 4-way capable Capacity-on-Demand model with only one module enabled, which activates both memory quads), both quads of memory are active. When both are active, it is important to note that the first and second modules are separate and independent. So this means that even though the size and speed of memory DIMMs behind each module have to be the same, the size and speed of memory DIMMs behind the first module do not have to match the memory DIMMs behind the second module. For DIMMs ranging from 512 MB to 4 GB, the speed is 667 Mbps (PC2-5300). The 8 GB DIMMs are different however, with a speed of 400 Mbps (PC2-3200). This decrease in speed for 8 GB DIMMs can have a negative effect on performance with memory sensitive workloads. This effect, along with the fact that there is no L3 cache, should be considered when planning for current and future growth and also LPAR configurations.

To test the performance difference of 4 GB DIMMs versus 8 GB DIMMs (essentially testing the difference in speed) and what occurs when the DIMMs of different sizes are "mixed", we used a Power6 520 (9408-M25) F/C 5635 (a fully enabled system) with one partition using all the available resources. "Mixed" here means the DIMMs in one quad behind a module are 4 GB and the DIMMs in the opposite quad are 8 GB. We started with a baseline consisting of all 4 GB DIMMs behind both modules, which is the best performing case. Then switched to all 8 GB DIMMs behind both modules and ran the same tests again. The performance of the workloads that were memory sensitive followed suit with the decrease in memory speed, which was expected. This is very important to consider when considering the amount of memory needed for a system. Deciding to go with the larger capacity 8 GB DIMMs does reduce your memory's speed and can have a negative performance effect on your workload. Of course each workload will behave differently based on its sensitivity to memory.

Next we placed 4 GB DIMMs behind one module and 8 GB DIMMs behind the opposite module. Because the one module had the faster 4 GB DIMMs behind it, the same workloads produced results that ranged between the best case, all 4 GB DIMMs, and the worst case, all 8 GB DIMMs. Again, we used only one partition that utilized all the available resources, but there are other factors to consider when using LPAR.

LPAR, or Logical Partitioning, increases flexibility, enabling selected system resources like processors, memory and I/O components to be utilized by various partitions, either in a shared or dedicated environment, on the same system. In the "mixed" environment previously described, it is possible to have one partition utilizing memory on 4 GB DIMMs and a second partition, configured with exactly the same amount of resources, utilizing memory on 8 GB DIMMs. This can cause an application to have different performance characteristics on the partitions. It is also possible for partitions to be assigned a mix of memory from different DIMMs, depending on how the memory is allocated at partition

activation time. This means that a partition that requires 4 GB of memory could be assigned 2 GB from the quad with 4 GB DIMMs and the other 2 GB from the quad with 8 GB DIMMs. This too can cause an application to have different performance characteristics on partitions configured with exactly the same amount of resources.

When system planning for the Power6 520, there are a number of memory related factors that should be considered, each of which can affect performance of memory sensitive workloads. First and foremost, the Power6 520 has no L3 cache. Having no L3 cache makes memory speed even more critical for memory sensitive workloads. If memory capacity needs can be achieved with 4 GB DIMMs or smaller, this will give the best memory speed. If memory capacity needs result in mixing 4 GB and 8 GB DIMMs, that option is available, but can have a negative performance effect on memory sensitive workloads. Mixing DIMMs can also cause partitions configured with exactly the same amount of resources to have varying performance characteristics. Since the Power6 520 only has 8 available memory DIMM slots, memory capacity can be an issue. If memory capacity is a concern, the 8 GB DIMMs will increase the capacity, but result in a slower memory speed.

## 20.6 Aligning Floating Point Data on Power6

The PowerPC architecture specifies that storage operands ought to be appropriately aligned. In many cases, there is a slight performance benefit and the compiler knows this, In other cases, the operands must be aligned for functional reasons. For example:

1. Pointers used by IBM i must be aligned on a 16-byte boundary,
2. PowerPC instructions in a program must be word aligned,
3. Binary Floating-Point operands ought to be word-aligned and should not cross a page boundary.

Other operand types allow generally free alignment of the data.

Although such a specification exists for Binary Floating-Point operands, the processor designs have the option of allowing free alignment of Binary Floating-Pointer operands as well. The Power6 processors, however, took a different approach. If either a 4-byte short form or 8-byte long form are not word-aligned, the Power6 processor will produce an alignment interrupt. Fortunately, the IBM i alignment interrupt handler recognizes this and does allow programs to successfully execute even if the Binary Floating-Point operand is not word aligned. However, this emulation of each such operation comes at a very considerable impact to the performance of such floating-point load and store instructions. While an appropriately aligned floating-point load or store can execute extremely rapidly, the emulation when misaligned can take thousands of times longer. If such accesses are rare compared to the remainder of the function being provided, this emulation may not matter to the performance of the application. As such floating-point accesses become more frequent, this emulation alone can account for most of the time spent within an application.

The compiler does attempt to assure that such Binary Floating-Point operands are at least word aligned. However, there are ways that the compiler's intent can be over-ridden. Packing data which includes floating-point variables within a structure may result in this occurring; packing of structures can occasionally save some space in memory. For this reason, it is prudent to assure that floating-point variables are allowed to be at least word aligned. If this can not be done, it may be appropriate to first copy the floating-point variables to a local aligned variable in storage; this may need to be done via an explicit move operation which is unaware of the type of the data for if the type is known; without this the

floating-point data may be copied using the floating-point loads and store, resulting in an alignment interrupt.

As an example, consider the following structures, one specifying "packed" and the other allowed to be aligned per the compiler. For example:

```
struct FPAlignmentStruct Packed
{
        long FloatingPointOp1;
        char ACharacter;
        long FloatingPointOp2; // Byte aligned; Can result in alignment interrupt.
}

struct FPAlignmentStructNormal          // Allows for preferred alignment
{
        long FloatingPointOp1;
        char ACharacter;
        long FloatingPointOp2; // Compiler padding added.
}
```

The first of these structures uses packing in order to minimize the amount of storage used. Here the structure consumes exactly 17 bytes, 8 each for the two floating-point values and one byte for the character. Assuming that the first is doubleword aligned as preferred, the second floating-point variable will be aligned on a doubleword+1 boundary. Each access of this second floating-point variable will result in an interrupt on Power6 processors.

The second of these structures allows the compiler to assure preferred alignment. Here the structure consumes exactly 24 bytes. The extra 7 bytes over the first comes from the compiler adding padding of seven bytes after the character variable in order to assure that the second floating-point variable is doubleword aligned.

If minimal storage is nonetheless required, there is another technique which will assure preferred alignment and minimal storage. This is accomplished by packaging the larger variables first as in the following example:

```
struct FPAlignmentStructNormal
{
        long FloatingPointOp1;
        long FloatingPointOp2; // Aligned without padding.
        char ACharacter;
}
```

This structure is also seventeen bytes in size and does assure preferred alignment.

# Chapter 21. High Availability Performance

The primary focus of this chapter is to present data that compares the effects of high availability scenarios using different hardware configurations. The data for the high availability test are broken down into two different categories which include Switchable IASP's, and Geographic Mirroring.

### High Availability Switchable Resources Considerations

Switchable IASPs are the physical resource that can be switched between systems in a cluster. A switchable IASP contains objects, the directories and libraries that contain the objects, and other object attributes such as authorization and ownership attributes.

Geographic Mirroring is a subfunction of cross-site mirroring (XSM) that generates a mirror image of an IASP on a system, which can be geographically distant from the originating site.

## 21.1 Switchable IASP's

There are three different switchover/failover scenarios that can occur in a switchable IASP environment.

Switchover: A cluster event where the primary database server or application server switches over to a backup system due to a manual intervention from the cluster management interface.

Failover: A cluster event where the primary database server or application server automatically switches to a backup system due to the failure of the primary server

Partition: A cluster event where communication is lost between one or more nodes in the cluster and a failure of the lost nodes cannot be confirmed. When a cluster partition condition is detected, cluster resource services limits the types of actions that you can perform on the nodes in the cluster partition.

NOTE: Failover performance is similar to switchover performance and therefore the workload was only run for switchover performance.

### Workload Description

Switchable IASP's using hardware resources

- Active Switchover - For an active switchover, the workload consists of bringing up a database workload on the IASP until the desired number of jobs are running on the system. Once the workload is stabilized the CHGCRGPRI(Change Cluster Resource Group Primary) command is issued from the command prompt. Switching time is measured from the time the CHGCRGPRI command is issued on the primary system until the new primary system's IASP is available. The CHGCRGPRI command ends all the jobs in the susbsystems that are using the IASP and thus time depends heavily on how many jobs are active on the IASP at the time the command is issued.

- Inactive switchover - The switching time is measured from the point at which the CHGCRGPRI command is issued from the primary system which has no work until the IASP is available on the new primary system.
- Partition - An active partition is created by starting the database workload on the IASP. Once the workload is stabilized an option 22(force MSD) is issued on the panel. Switching time is measured from the time the MSD is forced on the primary side until new primary node varies on the IASP.

## Workload Configuration

The wide variety of hardware configurations and software environments available make it difficult to characterize a 'typical' high availability environment. The following section provides a simple description of the high availability test environment used in our lab.

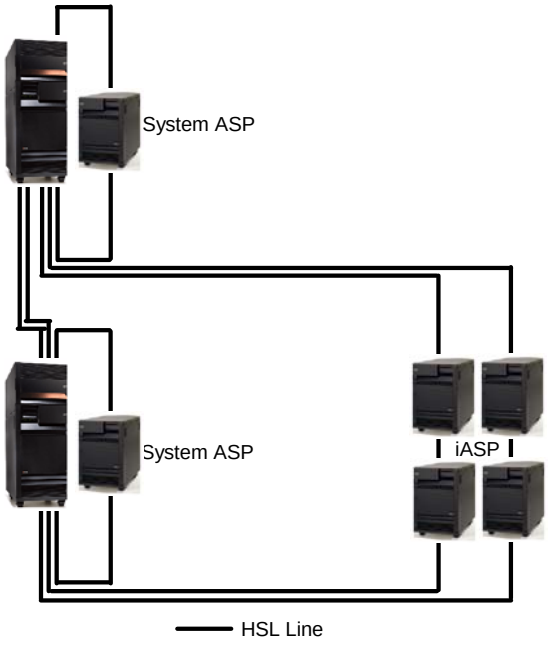
## System Configuration

Hardware Configuration

| | System A | System B |
|---|---|---|
| Model | 870 | 870 |
| Feature code | 7421 | 7421 |
| Interactive Code | 7421 | 7421 |
| Processor | 2486 | 2486 |
| CPW | 22000 | 22000 |
| # of Arms in System ASP | 50 | 50 |
| Speed of Dasd | 15k | 15k |
| Type of IOA's | 2757 | 2757 |
| Size of Dasd | 35 Gig | 35 Gig |
| Memory | 64Gig | 64 Gig |

| | iASP |
|---|---|
| # of Arms | 180 |
| Speed of Dasd | 15k |
| Type of IOA's | 2757 |
| Size of Dasd | 35 Gig |

Cabling Map




**870 Cluster HSL Cabling Map**

### Switchover Measurements

**NOTE:** The information that follows is based on performance measurements and analysis done in the Server Group Division laboratory. Actual performance may vary significantly from these tests.

### Switchable IASP's using Hardware Resources

*Time Required to Switch the IASP using Hardware Resources*

| | Inactive Switchovers | Active Switchovers | Active Partitions |
|---|---|---|---|
| Time(Minutes) | 4:31 | 10:19 | 6:55 |

### Switchover Tips

When planning an iSeries availability solution consider the characteristics of IASPs, as well as their advantages and disadvantages. For example, consider these statements regarding switched disks or IASPs when determining their value in an availability solution:

- For a faster vary on, keep the user-ID(UID) and group-ID(GID) of user profiles that own objects on the IASP the same between nodes of the cluster group. Having different UID's lengthens the vary on time significantly.
- The time to vary on an IASP during the switching process depends on the number of objects on the IASP, and not the size of the objects. If possible, keep the number of objects small.
- The number of devices in a tower affects switchover time. A larger number of devices in a switchable resource increases switchover time because devices must be reset.
- Keep the number of database objects in SYSBAS low on both systems. Larger number of objects in SYSBAS can slow the switchover.

---

## 21.2 Geographic Mirroring

A variety of scenarios exist in Cross-Site Mirroring that could be tested for performance. The best representative scenarios for the majority of our customers was measured.

With Geographic Mirroring we assessed the performance of the following components:

*Synchronization:* The geographic mirroring processing that copies data from the production copy to the mirror copy. During synchronization the mirror copy contains unusable data. When synchronization is completed, the mirror copy contains usable data.

Three resume priority options exist which may affect synchronization performance and time. Resume priority of low, medium, and high for geographic mirroring will affect the CPU utilization and the speed at which data is transferred. The default value is set at medium. A system set at high will transfer data faster and consume more CPU than a lower setting. Your choice will depend on how much time and CPU you want to allocate for this synchronization function.

*Switchable IASP's using Geographic Mirroring:* Refer to the switchable IASP's for a description of the different switchover scenarios that can occur using switchable IASP's with geographic mirroring.

*Active State:* In geographic mirroring, pertaining to the configuration state of a mirror copy that indicates geographic mirroring is being performed, if the IASP is online.

## Workload Description

Synchronization: This workload is performed by starting the synchronization process on the source side from an unsynchronized geographic mirrored IASP. The workload time is measured from the time geographic mirroring is activated on the source side until the target side has completed synchronization. Synchronization time is measured using the three resume priority levels of low, medium, and high.

Switchable IASPs using Geographic Mirroring:

- Active Switchover - The workload consists of bringing up a database workload on the IASP and letting it run until the desired number of jobs are active on the system. Once the workload is stabilized and the geographic mirror copy is synchronized the command is issued from the GUI or the CHGCRGPRI command to change the primary owner of the geographic mirrored copy. Switchover time is measured from the time the role change is issued from the GUI or the CHGCRGPRI until the new primary systems IASP is available.
- Inactive Switchover- Once the geographic mirror copy is synchronized the switchover is issued from the GUI or CHGCRGPRI command. Switchover time is measured from the time at which the switchover command is issued until the new primary systems IASP is available.
- Partition – After the geographic mirror copy is synchronized and the active workload is stabilized an option 22(force MSD) is issued on the panel. Switchover time is measured from the time the MSD is forced on the source side until the source node reports a failed status. After the failed status is reported the commands are issued to perform the switchover of the mirrored copy to a production copy.

Active State: The workload used was a slightly modified CPW workload for iASP environments. Initially a baseline without Geographic Mirroring is performed at 70% CPU utilization on a System/User ASP. The baseline value is then compared to various environment to assess the overhead of Geographic Mirroring.

## Workload Configuration

The wide variety of hardware configurations and software environments available make it difficult to characterize a 'typical' high availability environment and predict the results. The following section provides a simple description of the high availability test.

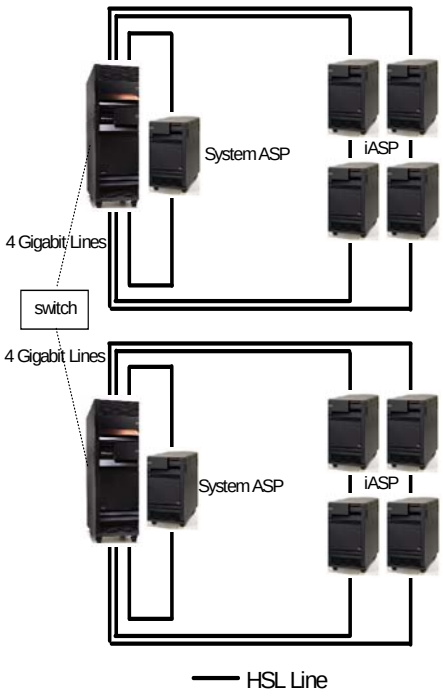
## Large System Configuration

Hardware Configuration

| | System A | System B |
|---|---|---|
| Model | 870 | 870 |
| Feature code | 7421 | 7421 |
| Interactive Code | 7421 | 7421 |
| Processor | 2486 | 2486 |
| CPW | 22000 | 22000 |
| # of Arms in System ASP | 50 | 50 |
| Speed of Dasd | 15k | 15k |
| Type of IOA's | 2757 | 2757 |
| Size of Dasd | 35 Gig | 35 Gig |
| Memory | 64Gig | 64 Gig |

| | iASP |
|---|---|
| # of Arms | 180 |
| Speed of Dasd | 15k |
| Type of IOA's | 2757 |
| Size of Dasd | 35 Gig |

Cabling Map




**870 Cluster HSL Cabling Map**

## Geographic Mirroring Measurements

**NOTE:** The information that follows is based on performance measurements and analysis done in the IBM Server Group Division laboratory. Actual performance may vary significantly from this test.

**Synchronization on an idle system:**

The following data shows the time required to synchronize 1 terabyte of data. This test case could vary greatly depending on the speed and latency of communication between the two systems. In the following measurements, the same switch was used for both the source and target systems. Also, large objects were synchronized were synchronized, which tend to synchronize faster than small objects.

*Time Required in Hours to Synchronize 1 Terabyte of Data using High Priority in Asynchronous mode*

| | 1 Gigabit Line | 2 Gigabit Lines | 3 Gigabit Lines | 4 Gigabit Lines |
|---|---|---|---|---|
| Time(Hours) | 2.75 | 1.4 | 1.25 | 1.1 |

*This case represents best case scenario. An environment with objects ¼ the size of the objects used in this test caused synchronization times 4x's larger.

*Effects of Resume Priority Settings on Synchronization of 1 Terabyte of data using 4 gigabit lines in Asynchronous Mode.*

| | Low | Medium | High |
|---|---|---|---|
| Time(Minutes) | 96:00 | 75:00 | 63:00 |
| Source System CPU Overhead | 9% | 12% | 16% |
| Target System CPU Overhead | 12% | 16% | 18% |

*This case represents best case scenario. An enviroment with objects ¼ the size of the objects used in this test caused synchronization times 4x's larger.

**Switchable Towers using Geographic Mirroring:**

The following data shows the time required to switch a geographic mirrored IASP that is synchronized from the source system to the target system.

*Time Required to Switch Towers using Geographic Mirroring using Asynchronous Mode*

| | Inactive Switchovers | Active Switchovers |
|---|---|---|
| Time(Minutes) | 6:00 | 6:00 |

**Active State:**

The following measurements show the CPU overhead from an System/User ASP baseline on the source system.

*CPU Overhead caused by Geographic Mirroring*

| | Asynchronous Geographic Mirroring Synchronization Stage | Asynchronous Geographic Mirroring | Synchronous Geographic Mirroring Synchronization Stage | Synchronous Geographic Mirroring |
|---|---|---|---|---|
| Source System CPU Overhead | 19% | 24% | 19% | 24% |
| Target System CPU Overhead | 13% | 13% | 13% | 13% |

- Geographic Mirroring Synchronization Stage: The number reflects the amount of CPU utilized while in the synchronization mode on a 1-line system.
- Asynchronous Geographic Mirroring: The number reflects the overhead caused by mirroring in asynchronous mode using 1-line system and running the CPW workload.
- Synchronous Geographic Mirroring: The number reflects overhead caused by mirroring in synchronous mode using 1-line and running the CPW workload

## Geographic Mirroring Tips

- For a quicker switchover time, keep the user-ID (UID) and group-ID (GID) of user profiles that own objects on the IASP the same between nodes of the cluster group. Having different UID's lengthens vary on times.
- Geographic mirroring is optimized for large files. A large number of small files will produce a slower synchronization rate.
- The priority settings available in the disk management section of iSeries navigator can improve the speed of the synchronization process. The tradeoff for faster speed however is a higher CPU utilization and could possibly degrade the applications running on the system during the synchronization process.
- Multiple TCP lines should be configured using TCP routes. Failure to use TCP routes will lead to a single line on the target side to be flooded with data. For more information look on the IBM InfoCenter.
- If geographic mirroring is not being used, geographic mirroring should not be configured. Configuring geographic mirroring without actually mirroring your data consumes up to 5% extra CPU.
- Increasing the number of lines will increase performance and reliability
- Place the journal in an IASP separate from the database to help the synchronization process

# Chapter 22. IBM Systems Workload Estimator

## 22.1 Overview

The IBM Systems Workload Estimator (a.k.a., the Estimator or WLE), located at: http://www.ibm.com/systems/support/tools/estimator, is a web-based sizing tool for System i, System p, and System x. You can use this tool to size a new system, to size an upgrade to an existing system, or to size a consolidation of several systems. The Workload Estimator allows measurement input to best reflect your current workload and provides a variety of built-in workloads to reflect your emerging application requirements. Virtualization can be reflected in the sizing to yield a more robust solution, by using various types of partitioning and virtual I/O. The Workload Estimator will provide current and growth recommendations for processor, memory, and disk that satisfy the overall customer performance requirements.

The Estimator supports sizings dealing with multiple systems, multiple partitions, multiple operating systems, and multiple time intervals. The Estimator also provides the ability to easily do multiple sizings. These features can be coordinated by using the functions on the Workload Selection screen.

The Estimator will recommend the system model including processor, memory, and DASD requirements that are necessary to handle the overall workload with reasonable performance expectations. In the case of System i5™, the Estimator may also recommend the 5250 OLTP feature or the Accelerator feature. To use the Estimator, you select one or more workloads and answer a few questions about each workload. Based on the answers, the Estimator generates a recommendation and shows the predicted CPU utilization of the recommended system in graphical format. The results can be viewed, printed, or generated in **P**ortable **D**ocument **F**ormat (PDF). The visualize solution function can be used to better understand the recommendation in terms of time intervals and virtualization. The Estimator can also be optionally linked to the System Planning Tool so that the configuration and validation may continue.

Sizing recommendations from the Estimator are based on processing capacity, which reflect the system's overall ability to handle the aggregate transaction rate. Again, this recommendation will yield processor, memory, and DASD requirements. Other aspects of sizing must also be considered beyond the scope of this tool. For example, to satisfy overnight batch windows or to deal with single-threaded applications, there may be additional unique hardware requirements that would allow adequate completion time. Also, you may need to increase the overall DASD recommendation to ensure that there is enough space to satisfy the overall storage requirements.

Sizing recommendations start with benchmarks and performance measurements based on well-defined, consistent workloads. For the built-in workloads in the Estimator, measurements have been done with numerous systems to characterize the workloads. Most of those workloads have parameters that allow them to be tailored to best suit the customer environment. This, again, is based on measurements and feedback from customers and Business Partners. Keep in mind, however, that many of these technologies are constantly evolving. IBM will continue to refine these workloads and sizing rules of thumb as IBM and our customers gain more experience.

As with every performance estimate (whether a rule of thumb or a sophisticated model), you always need to treat it as an estimate. This is particularly true with robust IBM systems that offer so many different capabilities where each installation will have unique performance characteristics and demands. The

typical disclaimers that go with any performance estimate ("your experience might vary...") are especially true. We provide these sizing estimates as general guidelines only.

## 22.2 Merging PM for System i data into the Estimator

The Measured Data workload of the Estimator is designed to accept data from various data sources. The most common ones are the PM for System i™ and PM for System p™. These are two tools that are tools available for the IBM System i™ and IBM System p™ respectively. These tools assist with many of the functions associated with capacity planning and performance analysis -- automatically. Either one will collect various data from your system that is critical to sizing and growth estimation. This data is then consolidated and sent to the Estimator. The Estimator will then use monthly, or weekly, statistics recorded from your system and show the system performance over time. The Estimator can use this information to more accurately determine the growth trends of the system workload.

The PM data is easily merged into the Estimator while viewing your PM graphs on the web. To view your PM iSeries graphs on the web, go to http://www.ibm.com/eserver/iseries/pm. Choose the 'click here to view your online reports' button.

Follow these instructions to merge the PM for System i data into the Estimator:
1. Enter your user id/password on the PM web site
2. Choose the 'Size my next upgrade' button.

Your PM data is then passed into the Estimator. If this is your first time using PM data with the Estimator, it is recommended that you take a few minutes to read the Measured Workload Integration tutorial, found on the help/tutorial tab in the Estimator.

## 22.3 Estimator Access

The intent is to provide a new version of the IBM Systems Workload Estimator 3 to 4 times per year. Each version includes an update message after approximately 3 months.

The IBM Systems Workload Estimator is available in two formats, on-line and as a download. Both are described in http://www.ibm.com/systems/support/tools/estimator. The on-line version is usually preferred.

It is also highly recommended that there should be involvement of IBM Sales or IBM Business Partners before making any purchasing decisions based on the results obtained from the Estimator.

The approximate size requirements are about 16.5 MB of hard disk space for the Workload Estimator and 60 MB for the server setup. A rough expectation of the time required to install the entire tool (server and Workload Estimator) is 20 minutes. A rough estimate of the time required for installing the update to Workload Estimator only, assuming the server was set up previously, is about 5 minutes.

## 22.4 What the Estimator is Not

The Estimator focuses on sizing based on capacity for processor, memory, and DASD. The Estimator does not recommend network adapters, communications media, I/O adapters, or configuration topology. The Estimator is not a configurator nor a configuration validation tool. The Estimator does not take into

account features like detailed journaling, resource locking, single-threaded applications, time-limited batch job windows, or poorly tuned environments.

The Estimator is a capacity sizing tool. Even though it does not represent actual transaction response times, it does adhere to the policy of giving recommendations that abide by generally accepted utilization thresholds. This means that the recommendation will likely have acceptable response times with the solution being prior to the knee of the curve on the common throughput vs. response time graph.

# Appendix A. CPW and CIW Descriptions

"Due to road conditions and driving habits, your results may vary." "Every workload is different." These are two hallmark statements of measuring performance in two very different industries. They are both absolutely correct. For iSeries and AS/400 systems, IBM has provided a measure called CPW to represent the relative computing power of these systems in a commercial environment. The type of caveats listed above are always included because no prediction can be made that a specific workload will perform in the same way that the workload used to generate CPW information performs.

Over time, IBM analysts have identified two sets of characteristics the appear to represent a large number of environments on iSeries and AS/400 systems. Many applications tend to follow the same patterns as CPW - which stands for **Commercial Processing Workload**. These applications tend to have many jobs running brief transactions in an environment that is dominated by IBM system code performing database operations. Other applications tend to follow the same patterns as CIW - which stands for **Compute Intensive Workload**. These applications tend to have somewhat fewer jobs running transactions which spend a substantial amount of time in the application, itself. The term "Compute Intensive" does not mean that commercial processing is not done. It simply means that more CPU power is typically expended in each transaction because more work is done at the application level instead of at the IBM licensed internal code level.

## A.1 Commercial Processing Workload - CPW

The CPW rating of a system is generated using measurements of a specific workload that is maintained internally within the iSeries Systems Performance group. CPW is designed to evaluate a computer system and associated software in the commercial environment. It is rigidly defined for function, performance metrics, and price/performance metrics. It is NOT representative of any specific environment, but it is generally applicable to the commercial computing environment.

- What CPW is
  - Test of a range of data base applications, including simple and medium complexity updates, simple and medium complexity inquiries, realistic user interfaces, and a combination of interactive and batch activities.
  - Test of commitment control
  - Test of concurrent data access by large numbers of users running a single group of programs.
  - Reasonable approximation of a steady-state, data base oriented commercial application.

- What CPW is not:
  - An indication of the performance capabilities of a system for any specific customer situation
  - A test of "ad-hoc" (query) data base performance

- When to use CPW data
  - Approximate product positioning between different AS/400 models where the primary application is expected to be oriented to traditional commercial business uses (order entry, payroll, billing, etc.) using commitment control

## CPW Application Description

The CPW application simulates the database server of an online transaction processing (OLTP) environment. Requests for transactions are received from an outside source and are processed by application service jobs on the database server. It is based, in part, on the business model from benchmarks owned and managed by the Transaction Processing Performance Council. However, there are substantive differences between this workload and public benchmarks that preclude drawing any correlation between them. For more information on public benchmarks from the Transaction Processing Performance Council, refer to their web page at www.tpc.org.

Specific choices were made in creating CPW to try to best represent the relative positioning of iSeries and AS/400 systems. Some of the differences between CPW and public benchmarks are:

- The code base for public benchmarks is constantly changing to try to obtain the best possible results, while an attempt is made to keep the base for CPW as constant as possible to better represent relative improvements from release to release and system to system.
- Public benchmarks typically do not require full security, but since IBM customers tend to run on secure systems, Security Level 50 is specified for the CPW workload
- Public benchmarks are super-tuned to obtain the best possible results for that specific benchmark, whereas for CPW we tend to use more of the system defaults to better represent the way the system is shipped to our customers.
- Public benchmarks can use different applications for different sized systems and take advantage of all of the resources available on a particular system, while CPW has been designed to run as the same application at all levels with approximately the same disk and memory resources per simulated user on all systems
- Public benchmarks tend to stress extreme levels of scaling at very high CPU utilizations for very limited applications. To avoid misrepresenting the capacity of larger systems, CPW is measured at approximately 70% CPU utilization.
- Public benchmarks require extensive, sophisticated driver and middle tier configurations. In order to simplify the environment and add a small computational component into the workload, CPW is driven by a batch driver that is included as a part of the overall workload.

The net result is an application that IBM believes provides an excellent indicator of transaction processing performance capacity when comparing between members of the iSeries and AS/400 families. As indicated above, CPW is not intended to be a guarantee of performance, but can be viewed as a good indicator.

The CPW application simulates the database server of an online transaction processing (OLTP) environment. There are five business functions of varying complexity that are simulated. These transactions are all executed by batch server jobs, although they could easily represent the type of transactions that might be done interactively in a customer environment. Each of the transactions interacts with 3-8 of the 9 database files that are defined for the workload. Database functions and file sizes vary. Functions exercised are single and multiple row retrieval, single and multiple row insert, single row update, single row delete, journal, and commitment control. These operations are executed against files that vary from 100's of rows to 100's of millions of rows. Some files have multiple indexes, some only one. Some accesses are to the actual data and some take advantage of advanced functions such as index-only access.

## A.2 Compute Intensive Workload - CIW

Unlike CPW values, CIW values are not derived from specific measurements of a single workload. They are modeled projections which are based upon the characteristics of internal workloads such as Domino workloads and application server environments such as can be found with SAP or JDEdwards applications. CIW is meant to depict a workload that has the following characteristics:

- The majority of the system procession time is spent in the user (or software supplier) application instead of system services. For example, a Domino Mail and Calendar workload might spend 80% of the total processing time outside of OS/400, while the CPW workload spends most of its time in OS/400 database code.
- Compute intensive applications tend to be considerably less I/O intensive than most commercial application processing. That is, more time is spent manipulating each piece of data than in a CPW-like environment.

- What CIW is
  - ❖ Indicator of relative performance in environments where a significant amount of transaction time is spent computing within the processor
  - ❖ Indicator of some of the differences between this type of workload and a "commercial" workload

- What CIW is not:
  - ❖ An indication of the performance capabilities of a system for any specific customer situation
  - ❖ A measure of pure numeric-intensive computing

- When to use CIW data
  - ❖ Approximate product positioning between different iSeries or AS/400 models where the primary application spends much of its time in the application code or middleware.

**What guidelines exist to help decide whether my workload is CIW-like or CPW-like?**
An absolute assignment of a workload is difficult without doing a very detailed analysis. The general rules listed here are probable placements, but not absolute guarantees. The importance of having the two measures is to show that different workloads react differently to changes in the configuration. IBM's Workload Estimator tries to take some of these differences into account when projecting how a workload will fit into a new system (see Appendix B.)

In general, if your application is online transaction processing (order entry, billing, accounts receivable, and the like), it will be CPW-like. If there are many, many jobs that spend more time waiting for a user to enter data than for the system to process it, it is likely to be CPW-like. If a significant part of the transaction response time is spent in disk and communications I/O, it is likely to be CPW-like. If the primary purpose of the application is to retrieve, process, and store database information, it is likely to be CPW-like.

CIW-like workloads tend to process less data with more instructions than CPW-like workloads. If your application is an "information manipulator" rather than an "information processor", it is probable that it will be CIW-like. This includes web-servers where much time is spent in generating and sending web frames and pages. It also includes application servers, where data is received from end-users, massaged and formatted into transaction requests, and then sent on to another system to actually service the database requests. If an application is both a "manipulator" and a "processor", experience has shown that enough time is spent in the manipulation portion of the application that it tends to be the dominant factor and the workload tends to be CIW-like. This is especially true of applications that are written using "modern" tools like Java, WebSphere Application Server, and WebSphere Commerce Suite. Another

category that often fits into the CIW-like classification is overnight batch. Even though batch jobs often process a great deal of database work, there are relatively few jobs which means there is little switching of jobs from processor to processor. As a result, overnight batch data processing jobs sometimes act more like compute-intensive jobs.

**What are the differences in how these workloads react to hardware configurations?**

When you upgrade your system, the effectiveness of the upgrade may be affected by the type of workload you are running. CPW-like workloads tend to respond well to upgrades in memory and to processor upgrades where the increase in MHz of the processor is accompanied by improvements in the processor cache and memory subsystem. CIW-like workloads tend to respond more to pure MHz improvements and to increasing the number of processors. You may experience both kinds of improvements. For example, there may be a difference between the way the daytime OLTP application reacts to an upgrade and the way the nighttime batch application reacts.

In a **CPW**-type workload, a lot of data is moved around and a wide variety of instructions are executed to manage the data. Because the transactions tend to be fairly short and because tasks are often waiting for new data to be brought from disk, processors are switched rapidly from task to task to task. This type of workload runs most efficiently when large amounts of the data it must process are readily available. Thus, it reacts favorably to large memory and large processor caches. We say that this type of workload is **cache-sensitive**. The bigger and faster the cache is, the more efficiently the workload runs (Note that cache is not an orderable feature. For iSeries, we attempt to balance processor upgrades with cache and memory subsystem upgrades whenever possible.) Increasing the MHz of the processor also helps, but you should not expect performance to scale directly with MHz unless other aspects of the system are equally improved. An example of this scenario can be found in V4R1, when the Model 640 systems were introduced as an upgrade path to Model 530 systems. The Model 640 systems actually had a lower MHz than the Model 530s, yet because they had much more cache and a much stronger memory implementation, they delivered a significantly higher CPW rating. Another aspect of CPW-type workloads is a dependency on a strong memory I/O subsystem. iSeries systems have always had a strong memory subsystem, but with the model 890, that subsystem was again significantly enhanced. Thus, CPW-like workloads see an additional benefit when moving to these systems.

In a **CIW**-type workload, the situation is somewhat different. Compute intensive workloads tend to processes less data with more instructions. As a consequence, the opportunity for both instruction and data cache hits is much higher in this kind of workload. Furthermore, because the instruction path length tends to be longer, it is likely that processors will switch from task to task much less often. Having some cache is very important for these workloads, but having a big cache is not nearly as important as it is for CPW-like workloads. For systems that are designed with enough cache and memory to accommodate CPW-like work, there is usually more than enough to assist CIW-like work and so an increase in MHz will tend to have a more dramatic effect on these workloads than it does on CPW-like work. CIW-like workloads tend to be **MHz-sensitive**. Furthermore, since tasks stay resident on individual processors longer, we tend to see better scaling on multiprocessor systems.

CPW and CIW ratings for iSeries systems can be found in Appendix D of this manual.

# Appendix B. System i Sizing and Performance Data Collection Tools

The following section presents some of the alternative tools available for sizing and capacity planning. (Note: There are products from vendors not included here that perform similar functions.) All of the tools discussed here support the current range of System i products, and include the capability to model logical partitions, partial processors (micropartitions) and server workload consolidation. The products supplied by vendors other than IBM require usage licenses from the respective vendor. All of these products depend on performance data collected through Collection Services.

- Performance Data Collection Services
  This tool which is part of the operating system collects system and job performance data which is the input for many of the sizing tools that are available today. It replaced the Performance Monitor in V5R1 and provides a more efficient and flexible way to collect performance data.

- IBM Systems Workload Estimator
  The IBM Systems Workload Estimator (a.k.a., the Estimator or WLE) is a web-based sizing tool for System i, System p, and System x. You can use this tool to size a new system, to size an upgrade to an existing system, or to size a consolidation of several systems. *See Chapter 22 for a discussion and a link for the IBM Systems Workload Estimator.*

- System i Batch Modeling tool STRBCHMDL. (BATCH400)
  This is best for MES upgrade sizing where the 'Batch Window' is important. BCHMDL uses Collection Services data to allow the user to view the batch jobs on a timeline. The elapsed time components (cpu, cpu queuing, disk, disk queuing, page faulting, etc.) are also available for viewing. The user can change the jobs or the configuration and run an analysis to determine the effect on batch runtime. The user can also model the effect of changing a single job into multiple jobs running concurrently. It can be found at: http://www.ibm.com/servers/eserver/iseries/perfmgmt/sizing.html

- PATROL for iSeries - Predict - BMC Software Inc
  Users of PATROL for iSeries – Predict interact through a 5250-interface with the performance database files to develop a capacity model for a system based on performance data collected during a period of peak utilization. The model is then downloaded to a PC for stand-alone predictive evaluation. The results show resource utilization and response times in reports and graphics. An enhancement supports LPAR configurations and the assignment of partial processors to partitions and attempts to predict the response time impact of virtual processors specifications. You will be able to find information about this product at *http://www.bmc.com*

- Performance Navigator - Midrange Performance Group Inc
  Sizing is one of the options included in the Performance Navigator product. Performance data can be collected on a regular (with the installation of Performance Navigator on the System i) or on an ad hoc basis. It can also use data prepared by PM for the System i platform. The sizing option is usually selected after evaluation of summary performance data, and represents a day of data. Continuous interaction with a System i host is required. A range of graphics present resource utilization of the system selected. The tool supports LPAR configurations and the assignment of partial processors to partitions. Performance Navigator presents a consolidated view of a multi-partition system. You will be able to find information about this product at http://www.mpginc.com

For more information on other System i Performance Tools, see the Performance Management web page at http://www.ibm.com/servers/eserver/iseries/perfmgmt/ .

## B.1 Performance Data Collection Services

Collecting performance data with Collection Services is an operating system function designed to run continuously that collects system and job level performance data at regular intervals which can be set from 15 seconds to 1 hour. It runs a number of collection routines called probes which collect data from many system resources including jobs, disk units, IOPs, buses, pools, and communication lines. Collection Services is the replacement for the Performance Monitor function which you may have used in previous releases to collect performance data by running the STRPFRMON command. Collection Services has been available in OS/400 since V4R4. The Performance Monitor remained on the system through V4R5 to allow time to switch over to the new Collection Services function.

### How Collection Services works

Collection Services has an improved method for storing the performance data that is collected. A system object called a management collection object (*MGTCOL) was created in V4R4 to store Collection Services data. The management collection object takes advantage of teraspace support to make it a more efficient way to store large quantities of performance data. Collection Services stores the data in a single collection object and supports a release independent design which allows you to move data to a system at a different release without requiring database file conversions.

A command called CRTPFRDTA (Create Performance Data) can be used to create the database files from the contents of the management collection object. The CRTPFRDTA command gives you the flexibility to generate only the database files you need to analyze a specific situation. If you decide that you always want to generate the database, you can configure Collection Services to run CRTPFRDTA as a low-priority batch job while data is being collected. Separating the collection of the data from the database generation, and running the database function at a lower priority are key reasons why Collection Services is efficient and can collect data from large quantities of jobs and threads at very frequent intervals. With Collection Services, you can collect performance data at intervals as frequent as every 15 seconds if you need that level of granularity to diagnose a performance problem. Collection Services also supports collection intervals of 30 seconds, and 1, 5, 15, 30, and 60 minutes.

The overhead associated with collecting performance data is minimal enough that Collection Services can run continuously, no matter what workload is being run on your system. If Collection Services is run continuously as designed, you will capture the data needed to analyze and solve many performance slowdowns before they turn into a serious problem.

### Starting Collection Services

You can start Collection Services by using option 2 on the Performance menu (GO PERFORM), the Collection Services component of iSeries Navigator, the STRPFRCOL command, or the QYPSSTRC (Start Collector) API. For more details on these options, see Performance under the Systems Management topic in the latest version of Information Center which is available at http://www.ibm.com/eserver/iseries/infocenter .

When using the Collection Services component of iSeries Navigator, you will find that it gives you flexibility to collect only the performance data you are interested in. Collection Services data is organized into over 20 categories and you have the ability to turn on and off each category or select a

predefined profile containing commonly used categories. For example, if you do not have a need to monitor the performance of SNADS transaction data on a regular basis, you can choose to turn that category off so that SNADS transaction data is not collected.

Since Collection Services is intended to be run continuously and trace mode is not, trace mode was not integrated into the start options of Collection Services. To run the trace mode facility you need to use two commands; STRPFRTRC (Start Performance Trace) and ENDPFRTRC (End Performance Trace). For more information on these commands, see Performance under the Systems Management topic in the latest Information Center which is available at http://www.ibm.com/eserver/iseries/infocenter.

---

## B.2 Batch Modeling Tool (BCHMDL).

BCHMDL is a tool for Batch Window Analysis available for recent systems. Instructions for requesting a copy are at the end of this description.

BCHMDL is a tool to enable System i batch window analysis to be done using information collected by Collection Services.

BCHMDL addresses the often asked question: 'What can I do to my system in order to meet my overnight batch run-time requirements (also known as the Batch Window).'

BCHMDL creates a 'model' from Collection Services performance data. This model will reside in a set of files named 'QAB4*' in the target library. The tool can then be asked to analyze the model and provide results for various 'what-if' conditions. Individual batch job run-time, and overall batch window run-times will be reported by this tool.

BCHMDL Output description:

1. Configuration summary shows the current and modeled hardware for DASD and CPU.

2. Job Statistics show the modeled results such as the following: elapsed time, cpu seconds, cpu queuing seconds (how long the job waited for the processor due to it being in use by other jobs), disk seconds, disk queuing, exceptional wait time, cpu %busy, etc.

3. Graph of Threads vs. Time of Day shows a 'horizontal' view of all threads in the model. This output is very handy in showing the relationship of job transitions within threads. It might indicate opportunities to break threads up to allow jobs to start earlier and run in parallel with jobs currently running in a sequential order.

4. Total CPU utilization shows a 'horizontal' view of how busy the CPU is. This report is on the same time-line as the previous Threads report.

After looking at the results, use the change option to make changes to the processor, disk, or to the jobs themselves. You can increase the total workload by making copies of jobs or by increasing the amount of work done by any given job. If you have a long running single threaded job, you could model how fast it would run as 4 multithreaded jobs by making 4 copies but make each job do 1/4th the work.

This tool will be available soon at:

http://www.ibm.com/servers/eserver/iseries/perfmgmt/batch.html

Unzip this file, transfer to your System i platform as a save file and restore library QBCHMDL. Add this library to your library list and start the tool by using the STRBCHMDL command. Tips, disclaimers, and general help are available in the QBCHMDL/README file. It is recommended that you work closely with your IBM Technical Support Representative when using this tool.

# Appendix C. CPW and MCU Relative Performance Values for System i

This chapter details the relative system performance values:

- Commercial Processing Workload (**CPW**). For a detailed description, refer to *Appendix A, "CPW Benchmark Description"*. CPW values are relative system performance metrics and reflect the relative system capacity for the CPW workload. CPW values can be used with caution in a capacity planning analysis (e.g., to scale CPU-constrained capacities, CPU time per transaction). However, these values may not appropriately reflect the performance of workloads other CPW because of differing detailed characteristics (e.g., cache miss ratios, average cycles per instruction, software contention, I/O characteristics, memory requirements, and application performance characteristics). The CPW values shown in the tables are based on IBM internal tests. Actual performance in a customer environment may vary significantly. Use the "IBM Systems Workload Estimator" for assistance with sizing; please refer to Chapter 22.

- Mail and Calendar Users (**MCU**). For a detailed description, refer to *Chapter 11, "Domino for System i"*. MCU values can be used to help size Domino environments for POWER5 and prior hardware. For new models, MCU values are not utilized or provided here.

- Compute Intensive Workload (**CIW**). For a detailed description, refer to *Appendix A.* CIW values are no longer utilized or provided here.

- User-based Licensing. Many newer models utilize user-based licensing for i5/OS. For assistance in determining the required number of user licenses, see the product web pages (for example: http://www.ibm.com/systems/i/hardware or http://www.ibm.com/systems/power/hardware) . Note that user-based licensing is not a performance statement or a replacement for system sizing; instead, user-based licensing only enables appropriate user connectivity to the system. Application environments differ in their requirements for system resources. Use the "IBM Systems Workload Estimator" for assistance with sizing based on performance.

- Relative Performance metric for System p (**rPerf**). System i systems that run AIX can be expected to produce the same performance as equivalent System p models given the same memory, disk, I/O, and workload configurations. The relative capacity of System p is often expressed in terms of rPerf values. The definition and the performance ratings for System p can be found at:
  - rPerf definition: http://www.ibm.com/systems/p/hardware/rperf.html
  - rPerf table: http://www.ibm.com/systems/p/hardware/system_perf.html

## C.1 V6R1 Additions (October 2008)

C.1.1 CPW values for the IBM Power Systems - IBM i operating system

***Table C.1.1.*** *CPW values for Power System Models*

| | | | | Processor CPW | | | | |
|---|---|---|---|---|---|---|---|---|
| Model | Processor Feature | Chip Speed GHz | L2/L3 cache (1) per chip | 2 cores | 4 cores | 8 cores | 12 cores | 16 cores |
| 570 (9117-MMA) | 7387 | 4.4 | 2x4MB / 32MB | 9850 | 19400 | 36200 | 51500 | 70000 |
| 570 (9117-MMA) | 7388 | 5.0 | 2x4MB / 32MB | 11000 | 21600 | 40300 | 56800 | 77600 |

*Note: 1. These models have a dedicated L2 cache per processor core, and share the L3 cache between two processor cores.
2. Memory speed differences account for some slight variations in performance difference between models.
3. CPW values for Power System models introduced in October 2008 were based on IBM i 6.1 plus enhancements in post-release PTFs.

C.1.2 CPW values for the IBM Power Systems - IBM i operating system

***Table C.1.2.*** *CPW values for Power System Models*

| | | | | Processor CPW | | | | |
|---|---|---|---|---|---|---|---|---|
| Model | Processor Feature | Chip Speed GHz | L2/L3 cache (1) per chip | 4 cores | 8 cores | 16 cores | 24 cores | 32 cores |
| 570 (9117-MMA) | 7540 | 4.2 | 2x4MB / 32MB | 16200 | 31900 | 56400 | 81600 | 104800 |

*Note: 1. These models have a dedicated L2 cache per processor core, and share the L3 cache between two processor cores.
2. Memory speed differences account for some slight variations in performance difference between models.
3. For large partitions, some workloads may experience nonlinear scaling at high system utilization on these new models.
4. CPW values for Power System models introduced in October 2008 were based on IBM i 6.1 plus enhancements in post-release PTFs.

C.1.3 CPW values for IBM Power Systems - IBM i operating system

***Table C.1.3.*** *CPW values for Power System Models*

| | | | | Processor CPW | | |
|---|---|---|---|---|---|---|
| Model | Processor Feature | Chip Speed GHz | L2/L3 cache (1) per chip | 4 cores | 8 cores | 16 cores |
| 560 (8234-EMA) | 7537 | 3.6 | 2x4MB / 32MB | 14100 | 27600 | 48500 |

*Note: 1. These models have a dedicated L2 cache per processor core, and share the L3 cache between two processor cores.

2. Memory speed differences account for some slight variations in performance difference between models.
3. CPW values for Power System models introduced in October 2008 were based on IBM i 6.1 plus enhancements in post-release PTFs.

### C.1.4 CPW values for IBM Power Systems - IBM i operating system

***Table C.1.4.*** *CPW values for Power System Models*

| Model | Processor Feature | Chip Speed GHz | L2/L3 cache (1) per chip | CPU (2) Range | Processor CPW |
|---|---|---|---|---|---|
| 520 (8203-E4A) | 5633 | 4.2 | 2x4MB / 0MB | 1 | 4300 |
| 520 (8203-E4A) | 5634 | 4.2 | 2x4MB / 0MB | 2 | 8300 |
| 520 (8203-E4A) | 5635 | 4.2 | 2x4MB / 0MB | 4 | 15600 |
| 550 (8204-E8A) | 4965 | 3.5 | 2x4MB / 32MB | 2 - 8 | 7750-27600 |
| 550 (8204-E8A) | 4966 | 4.2 | 2x4MB / 32MB | 2 - 8 | 9200-32650 |

*Note: 1. These models have a dedicated L2 cache per processor core, and share the L3 cache between two processor cores.
2. The range of the number of processor cores per system.
3. Memory speed differences account for some slight variations in performance difference between models.
4. CPW values for Power System models introduced in October 2008 were based on IBM i 6.1 plus enhancements in post-release PTFs.

## C.2 V6R1 Additions (August 2008)

### C.2.1 CPW values for the IBM Power 595 - IBM i operating system

***Table C.2.1.*** *CPW values for Power System Models*

| | | | | Processor CPW | | | | |
|---|---|---|---|---|---|---|---|---|
| Model | Processor Feature | Chip Speed MHz | L2/L3 cache (1) per chip | 8 cores | 16 cores | 24 cores | 32 cores | 64 cores (2) (2x32) |
| 595 (9119-FHA) | 4695 | 5000 | 2x4MB / 32MB | 41000 | 77000 | 108100 | 147900 | 294700 |
| 595 (9119-FHA) | 4694 | 4200 | 2x4MB / 32MB | 35500 | 66400 | 93800 | 128000 | 256200 |

*Note: 1. These models have a dedicated L2 cache per processor core, and share the L3 cache between two processor cores.
2. This configuration was measured with two 32-core partitions running simultaneously on a 64 core system

## C.3 V6R1 Additions (April 2008)

### C.3.1 CPW values for IBM Power Systems - IBM i operating system

| *Table C.3.1. CPW values for Power System Models* | | | | | |
|---|---|---|---|---|---|
| Model | Processor Feature | Chip Speed MHz | L2/L3 cache (1) per chip | CPU (2) Range | Processor CPW |
| 520 (9407-M15) | 5633 | 4200 | 2x4MB / 0MB | 1 | 4300 |
| 520 (9408-M25) | 5634 | 4200 | 2x4MB / 0MB | 1 - 2 | 4300-8300 |
| 550 (9409-M50) | 4966 | 4200 | 2x4MB / 32MB | 1 - 4 | 4800-18000 |

*Note: 1. These models have a dedicated L2 cache per processor core, and share the L3 cache between two processor cores.
2. The range of the number of processor cores per system.

C.3.2 CPW values for IBM BladeCenter JS12 - IBM i operating system

| *Table C.3.2. IBM BladeCenter models* | | | | | |
|---|---|---|---|---|---|
| Blade Model | Processor Feature | Chip Speed MHz | L2/L3 cache (1) per chip | CPUs(2) | Processor CPW (3) |
| JS12 (7998-60X) | 52BF | 3800 | 2x4MB / 0 MB | 1.8 of 2 | 7100 |

*Note: 1. These models have a dedicated L2 cache per processor core, and no L3 cache
2. CPW value is for a 1.8-core partition with shared processors and a 0.2-core VIOS partition
3. The value listed is unconstrained CPW (there is sufficient I/O such that the processor would be the first constrained resource). The I/O constrained CPW value for a 12-disk configuration is approximately 1200 CPW (100 CPW per disk).

C.3.3 CPW values for IBM Power Systems - IBM i operating system

| *Table C.3.3. CPW values for Power System Models* | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | Processor CPW | | | |
| Model | Processor Feature | Chip Speed MHz | L2/L3 cache (1) per chip | 2 cores | 4 cores | 8 cores | 16 cores |
| 570 (9117-MMA) | 5620 | 3500 | 2x4MB / 32MB | 8150 | 16100 | 30100 | 57600 |
| 570 (9117-MMA) | 5621/5622 | 4200 | 2x4MB / 32MB | 9650 | 19200 | 35500 | 68600 |
| 570 (9117-MMA) | 7380 | 4700 | 2x4MB / 32MB | 10800 | 21200 | 40100 | 76900 |

*Note: 1. These models have a dedicated L2 cache per processor core, and share the L3 cache between two processor cores.

## C.4 V6R1 Additions (January 2008)

C.4.1 IBM i5/OS running on IBM BladeCenter JS22 using POWER6 processor technology

| *Table C.4.1. IBM BladeCenter models* | | | | | | | |
|---|---|---|---|---|---|---|---|
| Blade Model | Server Feature | Edition Feature | Processor Feature | Chip Speed MHz | L2/L3 cache (1) per chip | CPUs | Processor CPW |
| JS22 (7998-61X) | n/a | n/a | 52BE | 4000 | 2x4MB / 0 MB | 3 of 4 (2) | 11040 |
| JS22 (7998-61X) | n/a | n/a | 52BE | 4000 | 2x4MB / 0 MB | 3.7 of 4 (3) | 13800 |

*Note: 1. These models have a dedicated L2 cache per processor core, and no L3 cache
2. CPW value is for a 3-core dedicated partition and a 1-core VIOS
3. CPW value is for a 3.7-core partition with shared processors and a 0.3-core VIOS partition

## C.5 V5R4 Additions (July 2007)

### C.5.1 IBM System i using the POWER6 processor technology

| *Table C.5.1. System i models* | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Model | Server Feature | Edition Feature[2] | Processor Feature | Chip Speed MHz | L2/L3 cache (1) per chip | CPU (5) Range | Processor CPW | MCU(4) |
| i570 (9406-MMA) | 4910 | 5460 | 7380 | 4700 | 2x4MB / 32MB | 1 - 4 | 5500-21200 | 12300-47500 |
| i570 (9406-MMA) | 4911 | 5461 | 7380 | 4700 | 2x4MB / 32MB | 2 - 8 | 10800-40100 | 24200-89700 |
| i570 (9406-MMA) | 4912 | 5462 | 7380 | 4700 | 2x4MB / 32MB | 4 - 16 | 20100-76900 | 45000-172000 |
| i570 (9406-MMA) | 4922 | 7053(3) | 7380 | 4700 | 2x4MB / 32MB | 1 - 4 | 5500-21200 | 12300-47500 |
| i570 (9406-MMA) | 4923 | 7058(3) | 7380 | 4700 | 2x4MB / 32MB | 1 - 8 | 5500-40100 | 12300-89700 |
| i570 (9406-MMA) | 4924 | 7063(3) | 7380 | 4700 | 2x4MB / 32MB | 2 - 16 | 10800-76900 | 24200-172000 |

*Note: 1. These models have a dedicated L2 cache per processor core, and share the L3 cache between two processor cores.
2. This is the Edition Feature for the model. This is the feature displayed when you display the system value QPRCFEAT.
3. Capacity Backup model.
4. Projected values. See Chapter 11 for more information.
5. The range of the number of processor cores per system.

## C.6 V5R4 Additions (January/May/August 2006 and January/April 2007)

### C.6.1 IBM System i using the POWER5 processor technology

| *Table C.6.1.1. System* i models | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Model | Edition Feature[2] | Accelerator Feature | Chip Speed MHz | L2/L3 cache per CPU (1) | CPU Range | Processor CPW | 5250 OLTP CPW | MCU |
| 9406-595 | 5892 | NA | 2300 | 1.9/36MB | 32 - 64 (8) | 108000-216000 | Per Processor | 242K(7)- 460K(7) |
| 9406-595 | 5872 | NA | 2300 | 1.9/36MB | 32 - 64 (8) | 108000-216000 | 0 | 242K(7)- 460K(7) |
| 9406-595 | 5891 | NA | 2300 | 1.9/36MB | 16 - 32 | 61000-108000 | Per Processor | 131K(7)- 242K(7) |
| 9406-595 | 5871 | NA | 2300 | 1.9/36MB | 16 - 32 | 61000-108000 | 0 | 131K(7)- 242K(7) |
| 9406-595 | 5896(4) | NA | 2300 | 1.9/36MB | 4 - 32 | 16000-108000 | Per Processor | 35800(7)- 242K(7) |
| 9406-595 | 5876(4) | NA | 2300 | 1.9/36MB | 4 - 32 | 16000-108000 | 0 | 35800(7)- 242K(7) |
| 9406-595 | 5890 | NA | 2300 | 1.9/36MB | 8-16 | 31500-58800 | Per Processor | 68400(7)- 131K(7) |

| ***Table C.6.1.1.*** *System* i models | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Model | Edition Feature[2] | Accelerator Feature | Chip Speed MHz | L2/L3 cache per CPU (1) | CPU Range | Processor CPW | 5250 OLTP CPW | MCU |
| 9406-595 | 5870 | NA | 2300 | 1.9/36MB | 8-16 | 31500-58800 | 0 | 68400(7)- 131K(7) |
| 9406-595 | 5895(4) | NA | 2300 | 1.9/36MB | 2-16 | 8200-58800 | Per Processor | 18300(7)- 131K(7) |
| 9406-595 | 5875(4) | NA | 2300 | 1.9/36MB | 2-16 | 8200-58800 | 0 | 18300(7)- 131K(7) |
| | | | | | | | | |
| 9406-595 | 7583(5) | NA | 1900 | 1.9/36MB | 32 - 64 (8) | 92000-184000 | Per Processor | 213K(7)- 405K(7) |
| 9406-595 | 7487 | NA | 1900 | 1.9/36MB | 32 - 64 (8) | 92000-184000 | Per Processor | 213K(7)- 405K(7) |
| 9406-595 | 7486 | NA | 1900 | 1.9/36MB | 32 - 64 (8) | 92000-184000 | 0 | 213K(7)- 405K(7) |
| 9406-595 | 7581(5) | NA | 1900 | 1.9/36MB | 16 - 32 | 51000-92000 | Per Processor | 115000 - 213K(7) |
| 9406-595 | 7483 | NA | 1900 | 1.9/36MB | 16 - 32 | 51000-92000 | Per Processor | 115000 - 213K(7) |
| 9406-595 | 7482 | NA | 1900 | 1.9/36MB | 16 - 32 | 51000-92000 | 0 | 115000 - 213K(7) |
| 9406-595 | 7590(4) | NA | 1900 | 1.9/36MB | 4 - 32 | 13600-92000 | Per Processor | 31500 - 213K(7) |
| 9406-595 | 7912(4) | NA | 1900 | 1.9/36MB | 4 - 32 | 13600-92000 | Per Processor | 31500 - 213K(7) |
| 9406-595 | 7580(5) | NA | 1900 | 1.9/36MB | 8 - 16 | 26700-50500 | Per Processor | 60500 - 114000 |
| 9406-595 | 7481 | NA | 1900 | 1.9/36MB | 8 - 16 | 26700-50500 | Per Processor | 60500 - 114000 |
| 9406-595 | 7480 | NA | 1900 | 1.9/36MB | 8 - 16 | 26700-50500 | 0 | 60500 - 114000 |
| 9406-595 | 7910(4) | NA | 1900 | 1.9/36MB | 2 - 16 | 6675-50500 | Per Processor | 15125 - 114000 |
| 9406-595 | 7911(4) | NA | 1900 | 1.9/36MB | 2 - 16 | 6675-50500 | Per Processor | 15125 - 114000 |
| | | | | | | | | |
| 9406-570 | 7760(4) | NA | 2200 | 1.9/36MB | 2 - 16 | 8100-58500 | Per Processor | 18200 - 130000 |
| 9406-570 | 7918(4) | NA | 2200 | 1.9/36MB | 2 - 16 | 8100-58500 | Per Processor | 18200 - 130000 |
| 9406-570 | 7765(5) | NA | 2200 | 1.9/36MB | 8 - 16 | 31100-58500 | Per Processor | 67500 - 130000 |
| 9406-570 | 7749 | NA | 2200 | 1.9/36MB | 8 - 16 | 31100-58500 | Per Processor | 67500 - 130000 |
| 9406-570 | 7759 | NA | 2200 | 1.9/36MB | 8 - 16 | 31100-58500 | 0 | 67500 - 130000 |
| 9406-570 | 7764(5) | NA | 2200 | 1.9/36MB | 4 - 8 | 16700-31100 | Per Processor | 35500 - 67500 |
| 9406-570 | 7748 | NA | 2200 | 1.9/36MB | 4 - 8 | 16700-31100 | Per Processor | 35500 - 67500 |
| 9406-570 | 7758 | NA | 2200 | 1.9/36MB | 4 - 8 | 16700-31100 | 0 | 35500 - 67500 |
| 9406-570 | 7916(4) | NA | 2200 | 1.9/36MB | 1 - 8 | 4200-31100 | Per Processor | 9100 - 67500 |
| 9406-570 | 7917(4) | NA | 2200 | 1.9/36MB | 1 - 8 | 4200-31100 | Per Processor | 9100 - 67500 |
| 9406-570 | 7763(5) | NA | 2200 | 1.9/36MB | 2 - 4 | 8400-16000 | Per Processor | 18200 - 34500 |
| 9406-570 | 7747 | NA | 2200 | 1.9/36MB | 2 - 4 | 8400-16000 | Per Processor | 18200 - 34500 |
| 9406-570 | 7757 | NA | 2200 | 1.9/36MB | 2 - 4 | 8400-16000 | 0 | 18200 - 34500 |
| 9406-570 | 7914(4) | NA | 2200 | 1.9/36MB | 1 - 4 | 4200-16000 | Per Processor | 9100 - 34500 |
| 9406-570 | 7915(4) | NA | 2200 | 1.9/36MB | 1 - 4 | 4200-16000 | Per Processor | 9100 - 34500 |
| | | | | | | | | |
| 9406-550 | 7551(5) | NA | 1900 | 1.9/36MB | 1 - 4 | 3800-14000 | Per Processor | 8200 - 30000 |
| 9406-550 | 7629(6) | NA | 1900 | 1.9/36MB | 1 - 4 | 3800-14000 | 0 | 8200 - 30000 |
| 9406-550 | 7155 | NA | 1900 | 1.9/36MB | 1 - 4 | 3800-14000 | Per Processor | 8200 - 30000 |
| 9406-550 | 7154 | NA | 1900 | 1.9/36MB | 1 - 4 | 3800-14000 | 0 | 8200 - 30000 |
| 9406-550 | 7920(4) | NA | 1900 | 1.9/36MB | 1 - 4 | 3800-14000 | Per Processor | 8200 - 30000 |
| 9406-550 | 7921(4) | NA | 1900 | 1.9/36MB | 1 - 4 | 3800-14000 | Per Processor | 8200 - 30000 |
| | | | | | | | | |
| 9406-525 | 7792(11) | NA | 1900 | 1.9/36MB | 1-2 | 3800-7100 | 3800-7100 | 8200 - 15600 |
| 9406-525 | 7791(11) | NA | 1900 | 1.9/36MB | 1-2 | 3800-7100 | 3800-7100 | 8200 - 15600 |
| 9406-525 | 7790(11) | NA | 1900 | 1.9/36MB | 1-2 | 3800-7100 | 3800-7100 | 8200 - 15600 |
| | | | | | | | | |
| 9407-515 | 6028(11) | NA | 1900 | 1.9/36MB | 2 | 7100(12) | 7100 | 15600(12) |
| 9407-515 | 6021(11) | NA | 1900 | 1.9/36MB | 2 | 7100(12) | 7100 | 15600(12) |
| 9407-515 | 6018(11) | NA | 1900 | 1.9/36MB | 1 | 3800(12) | 3800 | 8200(12) |
| 9407-515 | 6011(11) | NA | 1900 | 1.9/36MB | 1 | 3800(12) | 3800 | 8200(12) |
| 9407-515 | 6010(11) | NA | 1900 | 1.9/36MB | 1 | 3800(12) | 3800 | 8200(12) |
| | | | | | | | | |
| 9406-520 | 7375(5) | NA | 1900 | 1.9/36MB | 1 - 2 | 3800-7100 | 3800-7100 | 8200 - 15600 |
| 9406-520 | 7736 | NA | 1900 | 1.9/36MB | 1 - 2 | 3800-7100 | 3800-7100 | 8200 - 15600 |
| 9406-520 | 7785 | NA | 1900 | 1.9/36MB | 1 - 2 | 3800-7100 | 0 | 8200 - 15600 |
| 9406-520 | 7784 | NA | 1900 | 1.9/36MB | 1 | 3800 | 0 | 8200 |
| 9406-520 | 7691(10) | NA | 1900 | 1.9/36MB | 1 | 3800 | 0 | 8200 |
| 9406-520 | 7374(5) | NA | 1900 | 1.9/36MB | 1(3) | 2800 | 2800 | 6100 |
| 9406-520 | 7735 | NA | 1900 | 1.9/36MB | 1(3) | 2800 | 2800 | 6100 |

| *Table C.6.1.1. System* i models | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Model | Edition Feature[2] | Accelerator Feature | Chip Speed MHz | L2/L3 cache per CPU (1) | CPU Range | Processor CPW | 5250 OLTP CPW | MCU |
| 9406-520 | 7373(5) | NA | 1900 | 1.9/36MB | 1(3) | 1200 | 1200 | 2600 |
| 9406-520 | 7734 | NA | 1900 | 1.9/36MB | 1(3) | 1200 | 1200 | 2600 |
| **Value** | | | | | | | | |
| 9406-520 | 7352 | 7357 | 1900 | 1.9/36MB | 1(3) | 1200-3800 9 | 60 | 2600 - 8200 |
| 9406-520 | 7350 | 7355 | 1900 | 1.9MB/NA | 1(3) | 600-3100 9 | 30 | NR - 6600 |
| **Express** | | | | | | | | |
| 9405-520 | 7152 | NA | 1900 | 1.9/36MB | 1 | 3800 | 60 | 8200 |
| 9405-520 | 7144 | NA | 1900 | 1.9/36MB | 1 | 3800 | 60 | 8200 |
| 9405-520 | 7143 | 7354 | 1900 | 1.9/36MB | 1(3) | 1200-3800 9 | 60 | 2600 - 8200 (9) |
| 9405-520 | 7148 | 7687 | 1900 | 1.9/36MB | 1(3) | 1200-3800 9 | 60 | 2600 - 8200 (9) |
| 9405-520 | 7156 | 7353 | 1900 | 1.9/NA | 1(3) | 600-3100 9 | 30 | NR - 6600 (9) |
| 9405-520 | 7142 | 7682 | 1900 | 1.9MB/NA | 1(3) | 600-3100 9 | 30 | NR - 6600 (9) |
| 9405-520 | 7141 | 7681 | 1900 | 1.9MB/NA | 1(3) | 600-3100 9 | 30 | NR - 6600 (9) |
| 9405-520 | 7140 | 7680 | 1900 | 1.9MB/NA | 1(3) | 600-3100 9 | 30 | NR - 6600 (9) |

*Note: 1. These models share L2 and L3 cache between two processor cores.
2. This is the Edition Feature for the model. This is the feature displayed when you display the system value QPRCFEAT.
3. CPU Range - entry model is a partial processor model, offering multiple price/performance points for the entry market.
4. Capacity Backup model.
5. High Availability model.
6. Domino edition.
7. The MCU rating is a projected value.
8. The 64-way CPW value is reflects two 32-way partitions.
9. These models are accelerator models. The base CPW or MCU value is the capacity with the default processor feature. The max CPW or MCU value is the capacity when purchasing the accelerator processor feature.
10. Collaboration Edition. (Announced May 9, 2006)
11. User based pricing models.
12. These values listed are unconstrained CPW or MCU values (there is sufficient I/O such that the processor would be the first constrained resource). The I/O constrained CPW value for an 8-disk configuration is approximately 800 CPW (100 CPW per disk).

NR - Not Recommended: the 600 CPW processor offering is not recommended for Domino.

## C.7 V5R3 Additions (May, July, August, October 2004, July 2005)

New for this release is the eServer i5 servers which provide a significant performance improvement when compared to iSeries model 8xx servers.

### C.7.1 IBM @server® i5 Servers

| *Table C.7.1.1. @*server® i5 ***Servers*** | | | | | | | |
|---|---|---|---|---|---|---|---|
| Model | Chip Speed MHz | L2 cache per CPU (1) | L3 cache per CPU(2) | CPU Range | Processor CPW | 5250 OLTP CPW | MCU |
| 595-0952 (7485) | 1650 | 1.9 MB | 36 MB | 32 - 64 (8) | 86000-165000 | 12000-165000 | 196000(7)-375000(7) |
| 595-0952 (7484) | 1650 | 1.9 MB | 36 MB | 32 - 64 (8) | 86000-165000 | 0 | 196000(7)-375000(7) |
| 595-0947 (7499) | 1650 | 1.9 MB | 36 MB | 16 - 32 | 46000-85000 | 12000-85000 | 105000 -194000(7) |
| 595-0947 (7498) | 1650 | 1.9 MB | 36 MB | 16 - 32 | 46000-85000 | 0 | 105000 -194000(7) |

| *Table C.7.1.1. @server® i5 Servers* | | | | | | | |
|---|---|---|---|---|---|---|---|
| Model | Chip Speed MHz | L2 cache per CPU (1) | L3 cache per CPU(2) | CPU Range | Processor CPW | 5250 OLTP CPW | MCU |
| 595-0946 (7497) | 1650 | 1.9 MB | 36 MB | 8 - 16 | 24500-45500 | 12000-45500 | 54000-104000 |
| 595-0946 (7496) | 1650 | 1.9 MB | 36 MB | 8 - 16 | 24500-45500 | 0 | 54000-104000 |
| 570-0926 (7476) | 1650 | 1.9 MB | 36 MB | 13 - 16 | 36300-44700 | 12,000-44,700 | 83600-102000 |
| 570-0926 (7475) | 1650 | 1.9 MB | 36 MB | 13 - 16 | 36300-44700 | 0 | 83600-102000 |
| 570-0926 (7563)[5] | 1650 | 1.9 MB | 36 MB | 13 - 16 | 36300-44700 | 12000-44,700 | 83600-102000 |
| 570-0928 (7570)[4] | 1650 | 1.9 MB | 36 MB | 2 - 16 | 6350-44700 | 6,350-44,700 | 14100-102000 |
| 570-0928 (7474) | 1650 | 1.9 MB | 36 MB | 9 - 12 | 25500-33400 | 12,000-33,400 | 57300-77000 |
| 570-0924 (7473) | 1650 | 1.9 MB | 36 MB | 9 - 12 | 25500-33400 | 0 | 57300-77000 |
| 570-0924 (7562)[5] | 1650 | 1.9 MB | 36 MB | 9 - 12 | 25500-33400 | 12000-44,700 | 57300-77000 |
| 570-0922 (7472) | 1650 | 1.9 MB | 36 MB | 5 - 8 | 15200-23500 | 12,000-23,500 | 33600-52500 |
| 570-0922 (7471) | 1650 | 1.9 MB | 36 MB | 5 - 8 | 15200-23500 | 0 | 33600-52500 |
| 570-0922 (7561)[5] | 1650 | 1.9 MB | 36 MB | 5 - 8 | 15200-23500 | 12,000-23,500 | 33600-52500 |
| 570-0921 (7495) | 1650 | 1.9 MB | 36 MB | 2 - 4 | 6350-12000 | 12000 | 14100-26600 |
| 570-0921 (7494) | 1650 | 1.9 MB | 36 MB | 2 - 4 | 6350-12000 | 0 | 14100-26600 |
| 570-0921 (7560)[5] | 1650 | 1.9 MB | 36 MB | 2 - 4 | 6350-12000 | 12000 | 14100-26600 |
| 570-0930 (7491) | 1650 | 1.9 MB | 36 MB | 1 - 2 | 3300-6000 | 6000 | 7300-13300 |
| 570-0930 (7490) | 1650 | 1.9 MB | 36 MB | 1 - 2 | 3300-6000 | 0 | 7300-13300 |
| 570-0930 (7559)[5] | 1650 | 1.9 MB | 36 MB | 1 - 2 | 3300-6000 | 6,000 | 7300-13300 |
| 570-0920 (7470) | 1650 | 1.9 MB | 36 MB | 2 - 4 | 6350-12000 | Max | 14100-26600 |
| 570-0920 (7469) | 1650 | 1.9 MB | 36 MB | 2 - 4 | 6350-12000 | 0 | 14100-26600 |
| 570-0919 (7489) | 1650 | 1.9 MB | 36 MB | 1 - 2 | 3300-6000 | Max | 7300-13300 |
| 570-0919 (7488) | 1650 | 1.9 MB | 36 MB | 1 - 2 | 3300-6000 | 0 | 7300-13300 |
| 550-0915 (7530)[6] | 1650 | 1.9 MB | 36 MB | 2 - 4 | 6350-12000 | 0 | 14,100-26600 |
| 550-0915 (7463) | 1650 | 1.9 MB | 36 MB | 1 - 4 | 3300-12000 | 3,300-12,000 | 7300-26600 |
| 550-0915 (7462) | 1650 | 1.9 MB | 36 MB | 1 - 4 | 3300-12000 | 0 | 7300-26600 |
| 550-0915 (7558) | 1650 | 1.9 MB | 36 MB | 1 - 4 | 3300-12000 | 3,300-12,000 | 7300-26600 |
| 520-0905 (7457) | 1650 | 1.9 MB | 36 MB | 2 | 6000 | 3,300-6000 | 13300 |
| 520-0905 (7456) | 1650 | 1.9 MB | 36 MB | 2 | 6000 | 0 | 13300 |
| 520-0905 (7555)[5] | 1650 | 1.9 MB | 36 MB | 2 | 6000 | 3,300-6,000 | 13300 |
| 520-0904 (7455) | 1650 | 1.9 MB | 36 MB | 1 | 3300 | 3,300 | 7300 |
| 520-0904 (7454) | 1650 | 1.9 MB | 36 MB | 1 | 3300 | 0 | 7300 |
| 520-0904 (7554)[5] | 1650 | 1.9 MB | 36 MB | 1 | 3300 | 3,300 | 7300 |
| 520-0903 (7453) | 1500 | 1.9 MB | NA | 1 | 2400 | 2400 | 5500 |
| 520-0912 (7397) | 1500 | 1.9 MB | NA | 1 | 2400 | 60 | 5500 |
| 520-0912 (7395) | 1500 | 1.9 MB | NA | 1 | 2400 | 60 | 5500 |
| 520-0903 (7452) | 1500 | 1.9 MB | NA | 1 | 2400 | 0 | 5500 |
| 520-0903 (7553)[5] | 1500 | 1.9 MB | NA | 1 | 2400 | 2400 | 5500 |
| 520-0902 (7459) | 1500 | 1.9 MB | NA | 1 (3) | 1000 | 1000 | 2300 |
| 520-0902 (7458) | 1500 | 1.9 MB | NA | 1 (3) | 1000 | 0 | 2300 |
| 520-0902 (7552)[5] | 1500 | 1.9 MB | NA | 1 (3) | 1000 | 1000 | 2300 |
| 520-0901 (7451) | 1500 | 1.9MB | NA | 1 (3) | 1000 | 60 | 2300 |
| 520-0900 (7450) | 1500 | 1.9 MB | NA | 1 (3) | 500 | 30 | NA recommended |

*Note: 1. 1.9MB - These models share L2 cache between 2 processors.
2. 36MB - These models share L3 cache between 2 processors.
3. CPU Range - Partial processor models, offering multiple price/performance points for the entry market.
4. Capacity Backup model.
5. High Availability model.
6. Domino edition.
7. The MCU rating is a projected value.

8. The 64-way is measured as two 32-way partitions since i5/OS does not support a 64-way partition.
9. IBM stopped publishing CIW ratings for iSeries after V5R2. It is recommended that the IBM Systems Workload Estimator be used for sizing guidance, available at: http://www.ibm.com/eserver/iseries/support/estimator

## C.8 V5R2 Additions (February, May, July 2003)

New for this release is a product line refresh of the iSeries hardware which simplifies the model structure and minimizes the number of interactive choices. In most cases, the customer must choose between a Standard edition which includes a 5250 interactive CPW value of 0, or an Enterprise edition which supports the maximum 5250 OLTP capacity. The table in the following section lists the entire product line for 2003.

### C.8.1 iSeries Model 8xx Servers

***Table C.8.1.1. iSeries Models 8xx Servers***

| Model | Chip Speed MHz | L2 cache per CPU | CPU Range | Processor CPW | 5250 OLTP CPW* | Processor CIW* | MCU |
|---|---|---|---|---|---|---|---|
| 890-2498 (7427) | 1300 | 1.41 MB* | 24 - 32 | 29300-37400 | Max | 12900-16700 | 84100-108900 |
| 890-2498 (7425) | 1300 | 1.41 MB* | 24 - 32 | 29300-37400 | 0 | 12900-16700 | 84100-108900 |
| 890-2497 (7424) | 1300 | 1.41 MB* | 16 - 24 | 20000-29300 | Max | 8840-12900 | 57600-84100 |
| 890-2497 (7422) | 1300 | 1.41 MB* | 16 - 24 | 20000-29300 | 0 | 8840-12900 | 57600-84100 |
| 870-2486 (7421) | 1300 | 1.41 MB* | 8 - 16 | 11500-20000 | Max | 5280-9100 | 29600-57600 |
| 870-2486 (7419) | 1300 | 1.41 MB* | 8 - 16 | 11500-20000 | 0 | 5280-9100 | 29600-57600 |
| 870-2489 (7431) | 1300 | 1.41 MB* | 5 - 8 | 7700-11500 | 0 | 3600-5280 | 20200-29600 |
| 870-2489 (7433) | 1300 | 1.41 MB* | 5 - 8 | 7700-11500 | Max | 3600-5280 | 20200-29600 |
| 825-2473 (7418) | 1100 | 1.41 MB* | 3 - 6 | 3600-6600 | Max | 1570-2890 | 8700-17400 |
| 825-2473 (7416) | 1100 | 1.41 MB* | 3 - 6 | 3600-6600 | 0 | 1570-2890 | 8700-17400 |
| 810-2469 (7430) | 750 | 4 MB | 2 | 2700 | Max | 950 | 7900 |
| 810-2469 (7428) | 750 | 4 MB | 2 | 2700 | 0 | 950 | 7900 |
| 810-2467 (7412) | 750 | 4 MB | 1 | 1470 | Max | 530 | 4200 |
| 810-2467 (7410) | 750 | 4 MB | 1 | 1470 | 0 | 530 | 4200 |
| 810-2466 (7409) | 540 | 2 MB | 1 | 1020 | Max | 380 | 3100 |
| 810-2466 (7407) | 540 | 2 MB | 1 | 1020 | 0 | 380 | 3100 |
| 810-2465 (7406) | 540 | 2 MB | 1 | 750 | Max | 250 | 1900 |
| 810-2465 (7404) | 540 | 2 MB | 1 | 750 | 0 | 250 | 1900 |
| 800-2464 (7408) | 540 | 2 MB | 1 | 950 | 50 | 350 | 2900 |
| 800-2463 (7400) | 540 | 0 MB | 1 | 300 | 25 | - | - |

*Note: 1. 5250 OLTP CPW - Max (maximum CPW value). There is no limit on 5250 OLTP workloads and the full capacity of the server (Processor CPW) is available for 5250 OLTP work.
2. 1.41MB - These models share L2 cache between 2 processors
3. IBM does not intend to publish CIW ratings for iSeries after V5R2. It is recommended that the eServer Workload Estimator be used for sizing guidance, available at: http://www.ibm.com/eserver/iseries/support/estimator

## C.8.2 Model 810 and 825 iSeries for Domino (February 2003)

| *Table C.8.2.1. iSeries for Domino 8xx Servers* | | | | | | | |
|---|---|---|---|---|---|---|---|
| Model | Chip Speed MHz | L2 cache per CPU | CPU Range | Processor CPW | 5250 OLTP CPW* | Processor CIW* | MCU |
| | | | | | | | |
| 825-2473 (7416) | 1100 | 1.41 MB | 6 | 6600 | 0 | 2890 | 17400 |
| 825-2473 (7416) | 1100 | 1.41 MB | 4 | na | 0 | na | 11600 |
| | | | | | | | |
| 810-2469 (7428) | 750 | 4 MB | 2 | 2700 | 0 | 950 | 7900 |
| 810-2467 (7410) | 750 | 4 MB | 1 | 1470 | 0 | 530 | 4200 |
| 810-2466 (7407) | 540 | 2 MB | 1 | 1020 | 0 | 380 | 3100 |

*Note: 1. 5250 OLTP CPW - With a rating of 0, adequate interactive processing is available for a single 5250 job to perform system administration functions.
2. IBM does not intend to publish CIW ratings for iSeries after V5R2. It is recommended that the eServer Workload Estimator be used for sizing guidance, available at: http://www.ibm.com/eserver/iseries/support/estimator
na - indicates the rating is not available for the 4-way processor configuration

## C.9 V5R2 Additions

In V5R2 the following new iSeries models were introduced:
- 890 Base and Standard models
- 840 Base models
- 830 Base and Standard models

Base models represent server systems with "0" interactive capability. Standard Models represent systems that have interactive features available and also may have Capacity Upgrade on Demand Capability.

See Chapter 2, **iSeries RISC Server Model Performance Behavior,** for a description of the performance highlights of the new Dedicated servers for Domino models.

### C.9.1 Base Models 8xx Servers

| *Table C.9.1.1 Base Models 8xx Servers* | | | | | | | |
|---|---|---|---|---|---|---|---|
| Model | Chip Speed MHz | L2 cache per CPU | CPUs | Processor CPW | Interactive CPW | Processor CIW | MCU |
| 890-0198 (none) | 1300 | 1.41 MB* | 32 | 37400 | 0 | 16700 | 108900 |
| 890-0197 (none) | 1300 | 1.41 MB* | 24 | 29300 | 0 | 12900 | 84100 |
| 840-0159 (none) | 600 | 16 MB | 24 | 20200 | 0 | 10950 | 77800 |
| 840-0158 (none) | 600 | 16 MB | 12 | 12000 | 0 | 5700 | 40500 |
| 830-0153 (none) | 540 | 4 MB | 8 | 7350 | 0 | 3220 | 20910 |

* 890 Models share L2 cache between 2 processors

### C.9.2 Standard Models 8xx Servers

Standard models have an initial offering of processor and interactive capacity with featured upgrades for activation of additional processors and increased interactive capacity. Processor features are offered through Capacity Upgrade on Demand, described in ***C.10 V5R1 Additions***.

| *Table C.9.2.1 Standard Models 8xx Servers* | | | | | | | |
|---|---|---|---|---|---|---|---|
| Model | Chip Speed MHz | L2 cache per CPU | CPU Range | Processor CPW | Interactive CPW | Processor CIW | MCU |
| | | | | | | | |
| 890-2488 (1576) | 1300 | 1.41 MB* | 24 - 32 | 29300-37400 | 120 | 12900-16700 | 84100-108900 |
| 890-2488 (1577) | 1300 | 1.41 MB* | 24 - 32 | 29300-37400 | 240 | 12900-16700 | 84100-108900 |
| 890-2488 (1578) | 1300 | 1.41 MB* | 24 - 32 | 29300-37400 | 560 | 12900-16700 | 84100-108900 |
| 890-2488 (1579) | 1300 | 1.41 MB* | 24 - 32 | 29300-37400 | 1050 | 12900-16700 | 84100-108900 |
| 890-2488 (1581) | 1300 | 1.41 MB* | 24 - 32 | 29300-37400 | 2000 | 12900-16700 | 84100-108900 |
| 890-2488 (1583) | 1300 | 1.41 MB* | 24 - 32 | 29300-37400 | 4550 | 12900-16700 | 84100-108900 |
| 890-2488 (1585) | 1300 | 1.41 MB* | 24 - 32 | 29300-37400 | 10000 | 12900-16700 | 84100-108900 |
| 890-2488 (1587) | 1300 | 1.41 MB* | 24 - 32 | 29300-37400 | 16500 | 12900-16700 | 84100-108900 |
| 890-2488 (1588) | 1300 | 1.41 MB* | 24 - 32 | 29300-37400 | 20200 | 12900-16700 | 84100-108900 |
| 890-2488 (1591) | 1300 | 1.41 MB* | 24 - 32 | 29300-37400 | 37400 | 12900-16700 | 84100-108900 |
| | | | | | | | |
| 890-2487 (1576) | 1300 | 1.41 MB* | 16 - 24 | 20000-29300 | 120 | 8840-12900 | 57600-84100 |
| 890-2487 (1577) | 1300 | 1.41 MB* | 16 - 24 | 20000-29300 | 240 | 8840-12900 | 57600-84100 |
| 890-2487 (1578) | 1300 | 1.41 MB* | 16 - 24 | 20000-29300 | 560 | 8840-12900 | 57600-84100 |
| 890-2487 (1579) | 1300 | 1.41 MB* | 16 - 24 | 20000-29300 | 1050 | 8840-12900 | 57600-84100 |
| 890-2487 (1581) | 1300 | 1.41 MB* | 16 - 24 | 20000-29300 | 2000 | 8840-12900 | 57600-84100 |
| 890-2487 (1583) | 1300 | 1.41 MB* | 16 - 24 | 20000-29300 | 4550 | 8840-12900 | 57600-84100 |
| 890-2487 (1585) | 1300 | 1.41 MB* | 16 - 24 | 20000-29300 | 10000 | 8840-12900 | 57600-84100 |
| 890-2487 (1587) | 1300 | 1.41 MB* | 16 - 24 | 20000-29300 | 16500 | 8840-12900 | 57600-84100 |
| 890-2487 (1588) | 1300 | 1.41 MB* | 16 - 24 | 20000-29300 | 20200 | 8840-12900 | 57600-84100 |
| | | | | | | | |
| 830-2349 (1531) | 540 | 4 MB | 4 - 8 | 4200-7350 | 70 | 1630 - 3220 | 10680 - 20910 |
| 830-2349 (1532) | 540 | 4 MB | 4 - 8 | 4200-7350 | 120 | 1630 - 3220 | 10680 - 20910 |
| 830-2349 (1533 | 540 | 4 MB | 4 - 8 | 4200-7350 | 240 | 1630 - 3220 | 10680 - 20910 |
| 830-2349 (1534) | 540 | 4 MB | 4 - 8 | 4200-7350 | 560 | 1630 - 3220 | 10680 - 20910 |
| 830-2349 (1535) | 540 | 4 MB | 4 - 8 | 4200-7350 | 1050 | 1630 - 3220 | 10680 - 20910 |
| 830-2349 (1536) | 540 | 4 MB | 4 - 8 | 4200-7350 | 2000 | 1630 - 3220 | 10680 - 20910 |
| 830-2349 (1537) | 540 | 4 MB | 4 - 8 | 4200-7350 | 4550 | 1630 - 3220 | 10680 - 20910 |

* 890 Models share L2 cache between 2 processors

Other models available in V5R2 and listed in ***C.10 V5R1 Additions*** are as follows:
- All 270 Models
- All 820 Models
- Model 830-2400
- All 840 model listed in ***Table C.10.4.1.1 V5R1 Capacity Upgrade on-demand Models***

---

## C.10 V5R1 Additions

In V5R1 the following new iSeries models were introduced:
- 820 and 840 server models
- 270 server models
- 270 and 820 Dedicated servers for Domino
- 840 Capacity Upgrade on-demand models (including V4R5 models December 2000)

See Chapter 2, **iSeries RISC Server Model Performance Behavior,** for a description of the performance highlights of the new Dedicated Servers for Domino (DSD) models.

## C.10.1 Model 8xx Servers

| *Table C.9.1.1 Model 8xx Servers* | | | | | | | |
|---|---|---|---|---|---|---|---|
| Model | Chip Speed MHz | L2 cache per CPU | CPUs | Processor CPW | Interactive CPW | Processor CIW | MCU |
| 820-0150 (none) | 600 | 2 MB | 1 | 1100 | 0 | 385 | 3110 |
| 820-0151 (none) | 600 | 4 MB | 2 | 2350 | 0 | 840 | 6660 |
| 820-0152 (none) | 600 | 4 MB | 4 | 3700 | 0 | 1670 | 11810 |
| 820-2435 (1521) | 600 | 2 MB | 1 | 600 | 35 | 200 | 1620 |
| 820-2435 (1522) | 600 | 2 MB | 1 | 600 | 70 | 200 | 1620 |
| 820-2435 (1523) | 600 | 2 MB | 1 | 600 | 120 | 200 | 1620 |
| 820-2435 (1524) | 600 | 2 MB | 1 | 600 | 240 | 200 | 1620 |
| 820-2436 (1521) | 600 | 2 MB | 1 | 1100 | 35 | 385 | 3110 |
| 820-2436 (1522) | 600 | 2 MB | 1 | 1100 | 70 | 385 | 3110 |
| 820-2436 (1523) | 600 | 2 MB | 1 | 1100 | 120 | 385 | 3110 |
| 820-2436 (1524) | 600 | 2 MB | 1 | 1100 | 240 | 385 | 3110 |
| 820-2436 (1525) | 600 | 2 MB | 1 | 1100 | 560 | 385 | 3110 |
| 820-2437 (1521) | 600 | 4 MB | 2 | 2350 | 35 | 840 | 6660 |
| 820-2437 (1522) | 600 | 4 MB | 2 | 2350 | 70 | 840 | 6660 |
| 820-2437 (1523) | 600 | 4 MB | 2 | 2350 | 120 | 840 | 6660 |
| 820-2437 (1524) | 600 | 4 MB | 2 | 2350 | 240 | 840 | 6660 |
| 820-2437 (1525) | 600 | 4 MB | 2 | 2350 | 560 | 840 | 6660 |
| 820-2437 (1526) | 600 | 4 MB | 2 | 2350 | 1050 | 840 | 6660 |
| 820-2438 (1521) | 600 | 4 MB | 4 | 3700 | 35 | 1670 | 11810 |
| 820-2438 (1522) | 600 | 4 MB | 4 | 3700 | 70 | 1670 | 11810 |
| 820-2438 (1523) | 600 | 4 MB | 4 | 3700 | 120 | 1670 | 11810 |
| 820-2438 (1524) | 600 | 4 MB | 4 | 3700 | 240 | 1670 | 11810 |
| 820-2438 (1525) | 600 | 4 MB | 4 | 3700 | 560 | 1670 | 11810 |
| 820-2438 (1526) | 600 | 4 MB | 4 | 3700 | 1050 | 1670 | 11810 |
| 820-2438 (1527) | 600 | 4 MB | 4 | 3700 | 2000 | 1670 | 11810 |
| 830-2400 (1531) | 400 | 2 MB | 2 | 1850 | 70 | 580 | 4490 |
| 830-2400 (1532) | 400 | 2 MB | 2 | 1850 | 120 | 580 | 4490 |
| 830-2400 (1533) | 400 | 2 MB | 2 | 1850 | 240 | 580 | 4490 |
| 830-2400 (1534) | 400 | 2 MB | 2 | 1850 | 560 | 580 | 4490 |
| 830-2400 (1535) | 400 | 2 MB | 2 | 1850 | 1050 | 580 | 4490 |
| 830-2402 (1531) | 540 | 4 MB | 4 | 4200 | 70 | 1630 | 10680 |
| 830-2402 (1532) | 540 | 4 MB | 4 | 4200 | 120 | 1630 | 10680 |
| 830-2402 (1533) | 540 | 4 MB | 4 | 4200 | 240 | 1630 | 10680 |
| 830-2402 (1534) | 540 | 4 MB | 4 | 4200 | 560 | 1630 | 10680 |
| 830-2402 (1535) | 540 | 4 MB | 4 | 4200 | 1050 | 1630 | 10680 |
| 830-2402 (1536) | 540 | 4 MB | 4 | 4200 | 2000 | 1630 | 10680 |
| 830-2403 (1531) | 540 | 4 MB | 8 | 7350 | 70 | 3220 | 20910 |
| 830-2403 (1532) | 540 | 4 MB | 8 | 7350 | 120 | 3220 | 20910 |
| 830-2403 (1533) | 540 | 4 MB | 8 | 7350 | 240 | 3220 | 20910 |
| 830-2403 (1534) | 540 | 4 MB | 8 | 7350 | 560 | 3220 | 20910 |
| 830-2403 (1535) | 540 | 4 MB | 8 | 7350 | 1050 | 3220 | 20910 |
| 830-2403 (1536) | 540 | 4 MB | 8 | 7350 | 2000 | 3220 | 20910 |
| 830-2403 (1537) | 540 | 4 MB | 8 | 7350 | 4550 | 3220 | 20910 |
| 840-2461 (1540) | 600 | 16 MB | 24 | 20200 | 120 | 10950 | 77800 |
| 840-2461 (1541) | 600 | 16 MB | 24 | 20200 | 240 | 10950 | 77800 |
| 840-2461 (1542) | 600 | 16 MB | 24 | 20200 | 560 | 10950 | 77800 |
| 840-2461 (1543) | 600 | 16 MB | 24 | 20200 | 1050 | 10950 | 77800 |
| 840-2461 (1544) | 600 | 16 MB | 24 | 20200 | 2000 | 10950 | 77800 |

| *Table C.9.1.1 Model 8xx Servers* | | | | | | | |
|---|---|---|---|---|---|---|---|
| Model | Chip Speed MHz | L2 cache per CPU | CPUs | Processor CPW | Interactive CPW | Processor CIW | MCU |
| 840-2461 (1545) | 600 | 16 MB | 24 | 20200 | 4550 | 10950 | 77800 |
| 840-2461 (1546) | 600 | 16 MB | 24 | 20200 | 10000 | 10950 | 77800 |
| 840-2461 (1547) | 600 | 16 MB | 24 | 20200 | 16500 | 10950 | 77800 |
| 840-2461 (1548) | 600 | 16 MB | 24 | 20200 | 20200 | 10950 | 77800 |

Note: 830 models were first available in V4R5.

## C.10.2 Model 2xx Servers

| *Table C.10.2.1 Model 2xx Servers* | | | | | | | |
|---|---|---|---|---|---|---|---|
| Model | Chip Speed MHz | L2 cache per CPU | CPUs | Processor CPW | Interactive CPW | Processor CIW | MCU |
| 270-2431 (1518) | 540 | n/a | 1 | 465 | 30 | 185 | 1490 |
| 270-2432 (1516) | 540 | 2 MB | 1 | 1070 | 0 | 380 | 3070 |
| 270-2432 (1519) | 540 | 2 MB | 1 | 1070 | 50 | 380 | 3070 |
| 270-2434 (1516) | 600 | 4 MB | 2 | 2350 | 0 | 840 | 6660 |
| 270-2434 (1520) | 600 | 4 MB | 2 | 2350 | 70 | 840 | 6660 |

## C.10.3 V5R1 Dedicated Server for Domino

| *Table C.10.3 .1 Dedicated Servers for Domino* | | | | | | | |
|---|---|---|---|---|---|---|---|
| Model | Chip Speed MHz | L2 cache per CPU | CPUs | NonDomino CPW | Interactive CPW | Processor CIW | MCU |
| 270-2452 (none) | 540 | 2 MB | 1 | 100 | 0 | 380 | 3070 |
| 270-2454 (none) | 600 | 4 MB | 2 | 240 | 0 | 840 | 6660 |
| 820-2456 (none) | 600 | 2 MB | 1 | 120 | 0 | 385 | 3110 |
| 820-2457 (none) | 600 | 4 MB | 2 | 240 | 0 | 840 | 6660 |
| 820-2458 (none) | 600 | 4 MB | 4 | 380 | 0 | 1670 | 11810 |

### C.10.4 Capacity Upgrade on-demand Models

New in V4R5 (December 2000) , Capacity Upgrade on Demand (CUoD) capability offered for the iSeries Model 840 enables users to start small, then increase processing capacity without disrupting any of their current operations. To accomplish this, six processor features are available for the Model 840. These new processor features offer a Startup number of active processors; 8-way, 12-way or 18-way , with additional On-Demand processors capacity built-in (Standby). The customer can add capacity in increments of one processor (or more), up to the maximum number of On-Demand processors built into the Model 840. CUoD has significant value for installations who want to upgrade without disruption. To activate processors, the customer simply enters a unique activation code ("software key") at the server console (DST/SST screen).

The table below list the Capacity Upgrade on Demand features.

| | Startup Processors ("Active") | On-Demand Processors ("Stand-by") | TOTAL Processors |
|---|---|---|---|
| 840-2352 (2416) | 8 | 4 | 12 |
| 840-2353 (2417) | 12 | 6 | 18 |
| 840-2354 (2419) | 18 | 6 | 24 |

Note: Features 23xx added in V5R1. Features 24xx were available in V4R5 (December 2000)

#### C.10.4.1 CPW Values and Interactive Features for CUoD Models

The following tables list only the processor CPW value for the Startup number of processors as well as a processor CPW value that represents the full capacity of the server for all processors active (Startup + On-Demand). Interpolation between these values can give an approximate rating for incremental processor improvements, although the incremental improvements will vary by workload and because earlier activations may take advantage of caching resources that are shared among processors.

Interactive Features are available for the Model 840 ordered with CUoD Processor Features. Interactive performance is limited by total capacity of the active processors . When ordering FC 1546, FC 1547, or FC 1548 one should consider that the full capacity of interactive is not available unless all of the On-Demand processors have been activated .For more information on Capacity Upgrade on-demand, see URL: : http://www-1.ibm.com/servers/eserver/iseries/hardware/ondemand

Note: In V5R2, CUoD features come with all standard models, which are described in the **V5R2 Additions** section of this appendix.

| *Table C.10.4.1.1 V5R1 Capacity Upgrade on-demand Models* | | | | | | | |
|---|---|---|---|---|---|---|---|
| Model | Chip Speed MHz | L2 cache per CPU | CPU Range | Processor CPW | Interactive CPW | Processor CIW | MCU |
| 840-2352 (1540) | 600 | 16 MB | 8 - 12 | 9000 - 12000 | 120 | 3850 - 5700 | 27400 - 40500 |
| 840-2352 (1541) | 600 | 16 MB | 8 - 12 | 9000 - 12000 | 240 | 3850 - 5700 | 27400 - 40500 |
| 840-2352 (1542) | 600 | 16 MB | 8 - 12 | 9000 - 12000 | 560 | 3850 - 5700 | 27400 - 40500 |
| 840-2352 (1543) | 600 | 16 MB | 8 - 12 | 9000 - 12000 | 1050 | 3850 - 5700 | 27400 - 40500 |
| 840-2352 (1544) | 600 | 16 MB | 8 - 12 | 9000 - 12000 | 2000 | 3850 - 5700 | 27400 - 40500 |
| 840-2352 (1545) | 600 | 16 MB | 8 - 12 | 9000 - 12000 | 4550 | 3850 - 5700 | 27400 - 40500 |
| 840-2352 (1546) | 600 | 16 MB | 8 - 12 | 9000 - 12000 | 10000 | 3850 - 5700 | 27400 - 40500 |
| 840-2353 (1540) | 600 | 16 MB | 12 - 18 | 12000 - 16500 | 120 | 5700 - 8380 | 40500 - 59600 |
| 840-2353 (1541) | 600 | 16 MB | 12 - 18 | 12000 - 16500 | 240 | 5700 - 8380 | 40500 - 59600 |
| 840-2353 (1542) | 600 | 16 MB | 12 - 18 | 12000 - 16500 | 560 | 5700 - 8380 | 40500 - 59600 |
| 840-2353 (1543) | 600 | 16 MB | 12 - 18 | 12000 - 16500 | 1050 | 5700 - 8380 | 40500 - 59600 |
| 840-2353 (1544) | 600 | 16 MB | 12 - 18 | 12000 - 16500 | 2000 | 5700 - 8380 | 40500 - 59600 |
| 840-2353 (1545) | 600 | 16 MB | 12 - 18 | 12000 - 16500 | 4550 | 5700 - 8380 | 40500 - 59600 |
| 840-2353 (1546) | 600 | 16 MB | 12 - 18 | 12000 - 16500 | 10000 | 5700 - 8380 | 40500 - 59600 |
| 840-2353 (1547) | 600 | 16 MB | 12 - 18 | 12000 - 16500 | 16500 | 5700 - 8380 | 40500 - 59600 |
| 840-2354 (1540) | 600 | 16 MB | 18 - 24 | 16500 - 20200 | 120 | 8380 - 10950 | 59600 - 77800 |
| 840-2354 (1541) | 600 | 16 MB | 18 - 24 | 16500 - 20200 | 240 | 8380 - 10950 | 59600 - 77800 |
| 840-2354 (1542) | 600 | 16 MB | 18 - 24 | 16500 - 20200 | 560 | 8380 - 10950 | 59600 - 77800 |
| 840-2354 (1543) | 600 | 16 MB | 18 - 24 | 16500 - 20200 | 1050 | 8380 - 10950 | 59600 - 77800 |
| 840-2354 (1544) | 600 | 16 MB | 18 - 24 | 16500 - 20200 | 2000 | 8380 - 10950 | 59600 - 77800 |
| 840-2354 (1545) | 600 | 16 MB | 18 - 24 | 16500 - 20200 | 4550 | 8380 - 10950 | 59600 - 77800 |
| 840-2354 (1546) | 600 | 16 MB | 18 - 24 | 16500 - 20200 | 10000 | 8380 - 10950 | 59600 - 77800 |
| 840-2354 (1547) | 600 | 16 MB | 18 - 24 | 16500 - 20200 | 16500 | 8380 - 10950 | 59600 - 77800 |
| 840-2354 (1548) | 600 | 16 MB | 18 - 24 | 16500 - 20200 | 20200 | 8380 - 10950 | 59600 - 77800 |

| *Table C.10.4.1.2 V4R5 Capacity Upgrade on-demand Models (12/00)* | | | | | | | |
|---|---|---|---|---|---|---|---|
| Model | Chip Speed MHz | L2 cache per CPU | CPU Range | Processor CPW | Interactive CPW | Processor CIW | MCU |
| 840-2416 (1540) | 500 | 8 MB | 8 - 12 | 7800 - 10000 | 120 | 3100 - 4590 | 22000 - 32600 |
| 840-2416 (1541) | 500 | 8 MB | 8 - 12 | 7800 - 10000 | 240 | 3100 - 4590 | 22000 - 32600 |
| 840-2416 (1542) | 500 | 8 MB | 8 - 12 | 7800 - 10000 | 560 | 3100 - 4590 | 22000 - 32600 |
| 840-2416 (1543) | 500 | 8 MB | 8 - 12 | 7800 - 10000 | 1050 | 3100 - 4590 | 22000 - 32600 |
| 840-2416 (1544) | 500 | 8 MB | 8 - 12 | 7800 - 10000 | 2000 | 3100 - 4590 | 22000 - 32600 |
| 840-2416 (1545) | 500 | 8 MB | 8 - 12 | 7800 - 10000 | 4550 | 3100 - 4590 | 22000 - 32600 |
| 840-2416 (1546) | 500 | 8 MB | 8 - 12 | 7800 - 10000 | 10000 | 3100 - 4590 | 22000 - 32600 |
| 840-2417 (1540) | 500 | 8 MB | 12 - 18 | 10000 - 13200 | 120 | 4590 - 6750 | 32600 - 48000 |
| 840-2417 (1541) | 500 | 8 MB | 12 - 18 | 10000 - 13200 | 240 | 4590 - 6750 | 32600 - 48000 |
| 840-2417 (1542) | 500 | 8 MB | 12 - 18 | 10000 - 13200 | 560 | 4590 - 6750 | 32600 - 48000 |
| 840-2417 (1543) | 500 | 8 MB | 12 - 18 | 10000 - 13200 | 1050 | 4590 - 6750 | 32600 - 48000 |
| 840-2417 (1544) | 500 | 8 MB | 12 - 18 | 10000 - 13200 | 2000 | 4590 - 6750 | 32600 - 48000 |
| 840-2417 (1545) | 500 | 8 MB | 12 - 18 | 10000 - 13200 | 4550 | 4590 - 6750 | 32600 - 48000 |
| 840-2417 (1546) | 500 | 8 MB | 12 - 18 | 10000 - 13200 | 10000 | 4590 - 6750 | 32600 - 48000 |
| 840-2419 (1540) | 500 | 8 MB | 18 - 24 | 13200 - 16500 | 120 | 6750 - 8820 | 48000 - 62700 |
| 840-2419 (1541) | 500 | 8 MB | 18 - 24 | 13200 - 16500 | 240 | 6750 - 8820 | 48000 - 62700 |
| 840-2419 (1542) | 500 | 8 MB | 18 - 24 | 13200 - 16500 | 560 | 6750 - 8820 | 48000 - 62700 |
| 840-2419 (1543) | 500 | 8 MB | 18 - 24 | 13200 - 16500 | 1050 | 6750 - 8820 | 48000 - 62700 |
| 840-2419 (1544) | 500 | 8 MB | 18 - 24 | 13200 - 16500 | 2000 | 6750 - 8820 | 48000 - 62700 |
| 840-2419 (1545) | 500 | 8 MB | 18 - 24 | 13200 - 16500 | 4550 | 6750 - 8820 | 48000 - 62700 |
| 840-2419 (1546) | 500 | 8 MB | 18 - 24 | 13200 - 16500 | 10000 | 6750 - 8820 | 48000 - 62700 |
| 840-2419 (1547) | 500 | 8 MB | 18 - 24 | 13200 - 16500 | 16500 | 6750 - 8820 | 48000 - 62700 |

## C.11 V4R5 Additions

For the V4R5 hardware additions, the tables show each new server model characteristics and its maximum interactive CPW capacity. For previously existing hardware, the tables show for each server model the maximum interactive CPW and its corresponding CPU % and the point (the knee of the curve) where the interactive utilization begins to increasingly impact client/server performance. For the models that have multiple processors, and the knee of the curve is also given in CPU%, the percent value is the percent of all the processors (not of a single one).
CPW values may be increased as enhancements are made to the operating system (e.g. each feature of the Model 53S for V3R7 and V4R1). The server model behavior is fixed to the original CPW values.
For example, the model 53S-2157 had V3R7 CPWs of 509.9/30.7 and V4R1 CPWs 650.0/32.2. When using the 53S with V4R1, this means the knee of the curve is 2.6% CPU and the maximum interactive is 7.7% CPU, the same as it was in V3R7.

The 2xx, 8xx and SBx models are new in V4R5. See the chapter, **AS/400 RISC Server Model Performance Behavior,** for a description of the performance highlights of these new models.

### C.11.1 AS/400e Model 8xx Servers

***Table C.11.1 Model 8xx Servers (all new Condor models)***

| Model | Chip Speed MHz | L2 cache per CPU | CPUs | **Processor CPW** | **Interactive CPW** |
|---|---|---|---|---|---|
| 820-2395 (1521) | 400 | n/a | 1 | **370** | **35** |
| 820-2395 (1522) | 400 | n/a | 1 | **370** | **70** |
| 820-2395 (1523) | 400 | n/a | 1 | **370** | **120** |
| 820-2395 (1524) | 400 | n/a | 1 | **370** | **240** |
| 820-2396 (1521) | 450 | 2 MB | 1 | **950** | **35** |
| 820-2396 (1522) | 450 | 2 MB | 1 | **950** | **70** |
| 820-2396 (1523) | 450 | 2 MB | 1 | **950** | **120** |
| 820-2396 (1524) | 450 | 2 MB | 1 | **950** | **240** |
| 820-2396 (1525) | 450 | 2 MB | 1 | **950** | **560** |
| 820-2397 (1521) | 500 | 4 MB | 2 | **2000** | **35** |
| 820-2397 (1522) | 500 | 4 MB | 2 | **2000** | **70** |
| 820-2397 (1523) | 500 | 4 MB | 2 | **2000** | **120** |
| 820-2397 (1524) | 500 | 4 MB | 2 | **2000** | **240** |
| 820-2397 (1525) | 500 | 4 MB | 2 | **2000** | **560** |
| 820-2397 (1526) | 500 | 4 MB | 2 | **2000** | **1050** |
| 820-2398 (1521) | 500 | 4 MB | 4 | **3200** | **35** |
| 820-2398 (1522) | 500 | 4 MB | 4 | **3200** | **70** |
| 820-2398 (1523) | 500 | 4 MB | 4 | **3200** | **120** |
| 820-2398 (1524) | 500 | 4 MB | 4 | **3200** | **240** |
| 820-2398 (1525) | 500 | 4 MB | 4 | **3200** | **560** |
| 820-2398 (1526) | 500 | 4 MB | 4 | **3200** | **1050** |
| 820-2398 (1527) | 500 | 4 MB | 4 | **3200** | **2000** |
| 830-2400 (1531) | 400 | 2 MB | 2 | **1850** | **70** |
| 830-2400 (1532) | 400 | 2 MB | 2 | **1850** | **120** |
| 830-2400 (1533) | 400 | 2 MB | 2 | **1850** | **240** |
| 830-2400 (1534) | 400 | 2 MB | 2 | **1850** | **560** |
| 830-2400 (1535) | 400 | 2 MB | 2 | **1850** | **1050** |
| 830-2402 (1531) | 540 | 4 MB | 4 | **4200** | **70** |
| 830-2402 (1532) | 540 | 4 MB | 4 | **4200** | **120** |
| 830-2402 (1533) | 540 | 4 MB | 4 | **4200** | **240** |

| Model | Chip Speed MHz | L2 cache per CPU | CPUs | **Processor CPW** | **Interactive CPW** |
|---|---|---|---|---|---|
| 830-2402 (1534) | 540 | 4 MB | 4 | **4200** | **560** |
| 830-2402 (1535) | 540 | 4 MB | 4 | **4200** | **1050** |
| 830-2402 (1536) | 540 | 4 MB | 4 | **4200** | **2000** |
| | | | | | |
| 830-2403 (1531) | 540 | 4 MB | 8 | **7350** | **70** |
| 830-2403 (1532) | 540 | 4 MB | 8 | **7350** | **120** |
| 830-2403 (1533) | 540 | 4 MB | 8 | **7350** | **240** |
| 830-2403 (1534) | 540 | 4 MB | 8 | **7350** | **560** |
| 830-2403 (1535) | 540 | 4 MB | 8 | **7350** | **1050** |
| 830-2403 (1536) | 540 | 4 MB | 8 | **7350** | **2000** |
| 830-2403 (1537) | 540 | 4 MB | 8 | **7350** | **4550** |
| | | | | | |
| 840-2418 (1540) | 500 | 8 MB | 12 | **10000** | **120** |
| 840-2418 (1541) | 500 | 8 MB | 12 | **10000** | **240** |
| 840-2418 (1542) | 500 | 8 MB | 12 | **10000** | **560** |
| 840-2418 (1543) | 500 | 8 MB | 12 | **10000** | **1050** |
| 840-2418 (1544) | 500 | 8 MB | 12 | **10000** | **2000** |
| 840-2418 (1545) | 500 | 8 MB | 12 | **10000** | **4550** |
| 840-2418 (1546) | 500 | 8 MB | 12 | **10000** | **10000** |
| | | | | | |
| 840-2420 (1540) | 500 | 8 MB | 24 | **16500** | **120** |
| 840-2420 (1541) | 500 | 8 MB | 24 | **16500** | **240** |
| 840-2420 (1542) | 500 | 8 MB | 24 | **16500** | **560** |
| 840-2420 (1543) | 500 | 8 MB | 24 | **16500** | **1050** |
| 840-2420 (1544) | 500 | 8 MB | 24 | **16500** | **2000** |
| 840-2420 (1545) | 500 | 8 MB | 24 | **16500** | **4550** |
| 840-2420 (1546) | 500 | 8 MB | 24 | **16500** | **10000** |
| 840-2420 (1547) | 500 | 8 MB | 24 | **16500** | **16500** |

## C.11.2 Model 2xx Servers

***Table C.11.2.1 Model 2xx Servers***

| Model | Chip Speed MHz | L2 cache per CPU | CPUs | **Processor CPW** | **Interactive CPW** |
|---|---|---|---|---|---|
| 250-2295 | 200 | n/a | 1 | **50** | **15** |
| 250-2296 | 200 | n/a | 1 | **75** | **20** |
| | | | | | |
| 270-2248 (1517) | 400 | n/a | 1 | **150** | **25** |
| 270-2250 (1516) | 400 | n/a | 1 | **370** | **0** |
| 270-2250 (1518) | 400 | n/a | 1 | **370** | **30** |
| 270-2252 (1516) | 450 | 2 MB | 1 | **950** | **0** |
| 270-2252 (1519) | 450 | 2 MB | 1 | **950** | **50** |
| 270-2253 (1516) | 450 | 4 MB | 2 | **2000** | **0** |
| 270-2253 (1520) | 450 | 4 MB | 2 | **2000** | **70** |

## C.11.3 Dedicated Server for Domino

***Table C.11.3.1 Dedicated Server for Domino***

| Model | Chip Speed MHz | L2 cache per CPU | CPUs | **Non Domino CPW** | **Interactive CPW** |
|---|---|---|---|---|---|
| 820-2425 | 450 | 2 MB | 1 | **100** | **0** |
| 820-2426 | 500 | 4 MB | 2 | **200** | **0** |
| 820-2427 | 500 | 4 MB | 4 | **300** | **0** |
| | | | | | |
| 270-2422 | 400 | n/a | 1 | **50** | **0** |
| 270-2423 | 450 | 2 MB | 1 | **100** | **0** |
| 270-2424 | 450 | 4 MB | 2 | **200** | **0** |

## C.11.4 SB Models

| *Table C.11.4.1 SB Models* | | | | | |
|---|---|---|---|---|---|
| Model | Chip Speed MHz | L2 cache per CPU | CPUs | **Processor CPW*** | **Interactive CPW** |
| SB2-2315 | 540 | 4 MB | 8 | **7350** | **70** |
| SB3-2316 | 500 | 8 MB | 12 | **10000** | **120** |
| SB3-2318 | 500 | 8 MB | 24 | **16500** | **120** |

* Note: The "Processor CPW" values listed for the SB models are identical to the 830-2403-1531 (8-way), the 840-2418-1540 (12-way) and the 840-2420-1540 (24-way). However, due to the limited disk and memory of the SB models, it would not be possible to measure these values using the CPW workload. Disk space is not a high priority for middle-tier servers performing CPU-intensive work because they are always connected to another computer acting as the "database" server in a multi-tier implementation.

---

## C.12 V4R4 Additions

The Model 7xx is new in V4R4. Also in V4R4 are the Model 170s features 2289 and 2388 were added. See the chapter, **AS/400 RISC Server Model Performance Behavior,** for a description of the performance highlights of these new models.

Testing in the Rochester laboratory has shown that for systems executing traditional commercial applications such as RPG or COBOL interactive general business applications may experience about a 5% increase in CPU requirements. This effect was observed using the workload used to compute CPW, as shown in the tables that follows. Except for systems which are nearing the need for an upgrade, we do not expect this increase to significantly affect transaction response times. It is recommended that other sections of the Performance Capabilities Reference Manual (or other sizing and positioning documents) be used to estimate the impact of upgrading to the new release.

### C.12.1 AS/400e Model 7xx Servers

MAX Interactive CPW = Interactive CPW (Knee) * 7/6
CPU % used by Interactive @ Knee = Interactive CPW (Knee) / Processor CPW * 100
CPU % used by Processor @ Knee = 100 - CPU % used by Interactive @ Knee
CPU % used by Interactive @ Max = Max Interactive CPW / Processor CPW * 100

| *Table C.12.1.1 Model 7xx Servers (all new Northstar models)* | | | | | | |
|---|---|---|---|---|---|---|
| Model | Chip Speed MHz | L2 cache per CPU | CPUs | **Processor CPW** | **Interactive CPW (Knee)** | Interactive CPW (Max**)** |
| 720-2061 (Base) | 200 | n/a | 1 | **240** | **35** | 40.8 |
| 720-2061 (1501) | 200 | n/a | 1 | **240** | **70** | 81.7 |
| 720-2061 (1502) | 200 | n/a | 1 | **240** | **120** | 140 |
| | | | | | | |
| 720-2062 (Base) | 200 | 4 MB | 1 | **420** | **35** | 40.8 |
| 720-2062 (1501) | 200 | 4 MB | 1 | **420** | **70** | 81.7 |
| 720-2062 (1502) | 200 | 4 MB | 1 | **420** | **120** | 140 |
| 720-2062 (1503) | 200 | 4 MB | 1 | **420** | **240** | 280 |
| | | | | | | |
| 720-2063 (Base) | 200 | 4 MB | 2 | **810** | **35** | 40.8 |
| 720-2063 (1502) | 200 | 4 MB | 2 | **810** | **120** | 140 |
| 720-2063 (1503) | 200 | 4 MB | 2 | **810** | **240** | 280 |
| 720-2063 (1504) | 200 | 4 MB | 2 | **810** | **560** | 653.3 |
| | | | | | | |
| 720-2064 (Base) | 255 | 4 MB | 4 | **1600** | **35** | 40.8 |
| 720-2064 (1502) | 255 | 4 MB | 4 | **1600** | **120** | 140 |
| 720-2064 (1503) | 255 | 4 MB | 4 | **1600** | **240** | 280 |

| Model | Chip Speed MHz | L2 cache per CPU | CPUs | **Processor CPW** | **Interactive CPW (Knee)** | Interactive CPW (Max) |
|---|---|---|---|---|---|---|
| 720-2064 (1504) | 255 | 4 MB | 4 | **1600** | **560** | 653.3 |
| 720-2064 (1505) | 255 | 4 MB | 4 | **1600** | **1050** | 1225 |
| | | | | | | |
| 730-2065 (Base) | 262 | 4 MB | 1 | **560** | **70** | 81.7 |
| 730-2065 (1507) | 262 | 4 MB | 1 | **560** | **120** | 140 |
| 730-2065 (1508) | 262 | 4 MB | 1 | **560** | **240** | 280 |
| 730-2065 (1509) | 262 | 4 MB | 1 | **560** | **560** | 653.3 |
| | | | | | | |
| 730-2066 (Base) | 262 | 4 MB | 2 | **1050** | **70** | 81.7 |
| 730-2066 (1507) | 262 | 4 MB | 2 | **1050** | **120** | 140 |
| 730-2066 (1508) | 262 | 4 MB | 2 | **1050** | **240** | 280 |
| 730-2066 (1509) | 262 | 4 MB | 2 | **1050** | **560** | 653.3 |
| 730-2066 (1510) | 262 | 4 MB | 2 | **1050** | **1050** | 1225 |
| | | | | | | |
| 730-2067 (Base) | 262 | 4 MB | 4 | **2000** | **70** | 81.7 |
| 730-2067 (1508) | 262 | 4 MB | 4 | **2000** | **240** | 280 |
| 730-2067 (1509) | 262 | 4 MB | 4 | **2000** | **560** | 653.3 |
| 730-2067 (1510) | 262 | 4 MB | 4 | **2000** | **1050** | 1225 |
| 730-2067 (1511) | 262 | 4 MB | 4 | **2000** | **2000** | 2333.3 |
| | | | | | | |
| 730-2068 (Base) | 262 | 4 MB | 8 | **2890** | **70** | 81.7 |
| 730-2068 (1508) | 262 | 4 MB | 8 | **2890** | **240** | 280 |
| 730-2068 (1509) | 262 | 4 MB | 8 | **2890** | **560** | 653.3 |
| 730-2068 (1510) | 262 | 4 MB | 8 | **2890** | **1050** | 1225 |
| 730-2068 (1511) | 262 | 4 MB | 8 | **2890** | **2000** | 2333.3 |
| | | | | | | |
| 740-2069 (Base) | 262 | 8 MB | 8 | **3660** | **120** | 140 |
| 740-2069 (1510) | 262 | 8 MB | 8 | **3660** | **1050** | 1225 |
| 740-2069 (1511) | 262 | 8 MB | 8 | **3660** | **2000** | 2333.3 |
| 740-2069 (1512) | 262 | 8 MB | 8 | **3660** | **3660** | 4270 |
| | | | | | | |
| 740-2070 (Base) | 262 | 8 MB | 12 | **4550** | **120** | 140 |
| 740-2070 (1510) | 262 | 8 MB | 12 | **4550** | **1050** | 1225 |
| 740-2070 (1511) | 262 | 8 MB | 12 | **4550** | **2000** | 2333.3 |
| 740-2070 (1512) | 262 | 8 MB | 12 | **4550** | **3660** | 4270 |
| 740-2070 (1513) | 262 | 8 MB | 12 | **4550** | **4550** | 5308.3 |

## C.12.2 Model 170 Servers

### Current 170 Servers

MAX Interactive CPW = Interactive CPW (Knee) * 7/6
CPU % used by Interactive @ Knee = Interactive CPW (Knee) / Processor CPW * 100
CPU % used by Processor @ Knee = 100 - CPU % used by Interactive @ Knee
CPU % used by Interactive @ Max = Max Interactive CPW / Processor CPW * 100

***Table C.12.2.1 Current Model 170 Servers***

| Feature # | CPUs | Chip Speed | L2 cache per CPU | **Processor CPW** | **Interactive CPW (Knee)** | Interactive CPW (Max) | Processor CPU % @ Knee | Interactive CPU % @ Knee | Interactive CPU % @ Max |
|---|---|---|---|---|---|---|---|---|---|
| 2289 | 1 | 200 MHz | n/a | **50** | **15** | 17.5 | 70 | 30 | 35 |
| 2290 | 1 | 200 MHz | n/a | **73** | **20** | 23.3 | 72.6 | 27.4 | 32 |
| 2291 | 1 | 200 MHz | n/a | **115** | **25** | 29.2 | 78.3 | 21.7 | 25.4 |
| 2292 | 1 | 200 MHz | n/a | **220** | **30** | 35 | 86.4 | 13.6 | 15.9 |
| 2385 | 1 | 252 MHz | 4 MB | **460** | **50** | 58.3 | 89.1 | 10.9 | 12.7 |
| 2386 | 1 | 252 MHz | 4 MB | **460** | **70** | 81.7 | 84.8 | 15.2 | 17.8 |
| 2388 | 2 | 255 MHz | 4 MB | **1090** | **70** | 81.7 | 92.3 | 6.4 | 7.5 |

Note: the CPU not used by the interactive workloads at their Max CPW is used by the system CFINTnn jobs. For example, for the 2386 model the interactive workloads use 17.8% of the CPU at their maximum and the CFINTnn jobs use the remaining 82.2%. The processor workloads use 0% CPU when the interactive workloads are using their maximum value.

## AS/400e Dedicated Server for Domino

*Table C.12.2.2 Dedicated Server for Domino*

| Feature # | CPUs | Chip Speed | L2 cache per CPU | **Processor CPW** | **Interactive CPW** | Processor CPU% @ Knee | Processor CPU % @ Max | Interactive CPU % @ Knee | Interactive CPU % @ Max |
|---|---|---|---|---|---|---|---|---|---|
| 2407 | 1 | n/a | n/a | 30 | 10 | - | - | - | - |
| 2408 | 1 | n/a | 4 MB | 60 | 15 | - | - | - | - |
| 2409 | 2 | n/a | 4 MB | 120 | 20 | - | - | - | - |

## Previous Model 170 Servers

On previous Model 170's the knee of the curve is about 1/3 the maximum interactive CPW value.

Note that a constrained (c) CPW rating means the maximum memory or DASD configuration is the constraining factor, not the processor. An unconstrained (u) CPW rating means the processor is the first constrained resource.

*Table C.12.2.3 Previous Model 170 Servers*

| Feature # | Constrain / Unconstr | **Client / Server CPW** | **Interactive CPW (Max)** | Interactive CPW (Knee) | Interactive CPU % @ Max | Interactive CPU % @ Knee |
|---|---|---|---|---|---|---|
| 2159 | c | **73** | **16** | 5.3 | 22.2 | 7.7 |
| | u | **73** | **16** | 5.3 | 22.2 | 7.7 |
| 2160 | c | **114** | **23** | 7.7 | 21.2 | 7.4 |
| | u | **114** | **23** | 7.7 | 21.2 | 7.4 |
| 2164 | c | **125** | **29** | 9.7 | 14 | 4.7 |
| | u | **210** | **29** | 9.7 | 14 | 4.7 |
| 2176 | c | **125** | **40** | 13.3 | 12.9 | 4.4 |
| | u | **319** | **40** | 13.3 | 12.9 | 4.4 |
| 2183 | c | **125** | **67** | 22.3 | 21.5 | 7.2 |
| | u | **319** | **67** | 22.3 | 21.5 | 7.2 |

## C.13 AS/400e Model Sxx Servers

For AS/400e servers the knee of the curve is about 1/3 the maximum interactive CPW value.

*Table C.13.1 AS/400e Servers*

| Model | Feature # | CPUs | **Max C/S CPW** | **Max Inter CPW** | 1/3 Max Interact CPW | CPU % @ Max Interact | CPU % @ the Knee |
|---|---|---|---|---|---|---|---|
| S10 | 2118 | 1 | **45.4** | **16.2** | 5.4 | 35.7 | 11.9 |
| | 2119 | 1 | **73.1** | **24.4** | 8.1 | 33.4 | 11.1 |
| S20 | 2161 | 1 | **113.8** | **31** | 10.3 | 27.2 | 9.1 |
| | 2163 | 1 | **210** | **35.8** | 11.9 | 17 | 5.7 |
| | 2165 | 2 | **464.3** | **49.7** | 16.7 | 10.7 | 3.6 |
| | 2166 | 4 | **759** | **56.9** | 19.0 | 7.5 | 2.5 |
| S30 | 2257 | 1 | **319** | **51.5** | 17.2 | 16.1 | 5.4 |
| | 2258 | 2 | **583.3** | **64** | 21.3 | 11 | 3.7 |
| | 2259 | 4 | **998.6** | **64** | 21.3 | 6.4 | 2.1 |
| | 2260 | 8 | **1794** | **64** | 21.3 | 3.6 | 1.2 |
| S40 | 2207 | 8 | **3660** | **120** | 40 | 3.2 | 1.1 |
| | 2208 | 12 | **4550** | **120** | 40 | 2.6 | 0.8 |
| | 2256 | 8 | **1794** | **64** | 21.3 | 3.6 | 1.2 |
| | 2261 | 12 | **2340** | **64** | 21.3 | 2.7 | 0.9 |

## C.14 AS/400e Custom Servers

For custom servers the knee of the curve is about 6/7 maximum interactive CPW value.

*Table C.14.1 AS/400e Custom Servers*

| Model | Feature # | CPUs | **Max** | **Max** | 6/7 Max | CPU % @ | CPU % |
|---|---|---|---|---|---|---|---|
| S20 | 2177 | 4 | **759** | **110.7** | 94.9 | 14.6 | 12.5 |
| | 2178 | 4 | **759** | **221.4** | 189.8 | 29.2 | 25.0 |
| S30 | 2320 | 4 | **998.6** | **215.1** | 184.4 | 21.5 | 18.5 |
| | 2321 | 8 | **1794** | **386.4** | 331.2 | 21.5 | 18.5 |
| | 2322 | 8 | **1794** | **579.6** | 496.8 | 32.5 | 27.7 |
| S40 | 2340 | 8 | **3660** | **1050.0** | 900.0 | 28.6 | 24.5 |
| | 2341 | 12 | **4550** | **2050.0** | 1757.1 | 38.6 | 33.1 |

## C.15 AS/400 Advanced Servers

For AS/400 Advanced Servers the knee of the curve is about 1/3 the maximum interactive CPW value.

For releases prior to V4R1 the model 150 was constrained due to the memory capacity. With the larger capacity for V4R1, memory is no longer the limiting resource. In V4R1, the limit of 4 DASD devices is the constraining resource. For workloads that do not perform as many disk operations or don't require as much memory, the unconstrained CPW value may be more representative of the performance capabilities. An unconstrained CPW rating means the processor is the first constrained resource.

| *Table C.15.1 AS/400 Advanced Servers: V4R1 and V4R2* | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Model | Feature # | Constrain / Unconstr | CPUs | **Max C/S CPW** | **Max Inter CPW** | 1/3 Max Interact CPW | CPU % @ Max Interact | CPU % @ the Knee |
| 150 | 2269 | c | 1 | **20.2** | **13.8** | 4.6 | 51.1 | 17 |
| | 2269 | u | 1 | **27** | **13.8** | 4.6 | 51.1 | 17 |
| | 2270 | c | 1 | **20.2** | **20.2** | 6.7 | 61.9 | 20.6 |
| | 2270 | u | 1 | **35** | **20.6** | 6.9 | 61.9 | 20.6 |
| 40S | 2109 | n/a | 1 | **27** | **9.4** | 3.1 | 30.1 | 10 |
| | 2110 | n/a | 1 | **35** | **14.5** | 3.9 | 37.4 | 12.5 |
| 50S | 2111 | n/a | 1 | **63.0** | **21.6** | 7.2 | 29.8 | 9.9 |
| | 2112 | n/a | 1 | **91.0** | **32.2** | 10.8 | 29.8 | 9.9 |
| | 2120 | n/a | 1 | **81.6** | **22.5** | 8.1 | 27.8 | 9.3 |
| | 2121 | n/a | 1 | **111.5** | **32.2** | 10.7 | 30 | 10 |
| | 2122 | n/a | 1 | **138.0** | **32.2** | 12.0 | 23.8 | 8.9 |
| 53S | 2154 | n/a | 1 | **188.2** | **32.2** | 15.9 | 20.3 | 6.8 |
| | 2155 | n/a | 2 | **319.0** | **32.2** | 10.7 | 13.5 | 4.5 |
| | 2156 | n/a | 4 | **598.0** | **32.2** | 10.7 | 9 | 3 |
| | 2157 | n/a | 4 | **650.0** | **32.2** | 10.9 | 7.7 | 2.6 |

| *Table C.15.2 AS/400 Advanced Servers: V3R7* | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Model | Feature # | Constrain / Unconstr | CPUs | **Max C/S CPW** | **Max Inter CPW** | 1/3 Max Interact CPW | CPU % @ Max Interact | CPU % @ the Knee |
| 150 | 2269 | c | 1 | **10.9** | **10.9** | 3.6 | 100.0 | 33.0 |
| | 2269 | u | 1 | **10.9** | **10.9** | 3.6 | 100.0 | 33.0 |
| | 2270 | c | 1 | **27.0** | **13.8** | 4.6 | 51.1 | 17.0 |
| | 2270 | u | 1 | **33.3** | **20.6** | 6.9 | 61.9 | 20.6 |
| 40S | 2109 | n/a | 1 | **27.0** | **9.4** | 3.1 | 30.1 | 10 |
| | 2110 | n/a | 1 | **33.3** | **13.8** | 3.7 | 37.4 | 12.5 |
| | 2111 | n/a | 1 | **59.8** | **20.6** | 6.9 | 29.8 | 9.9 |
| | 2112 | n/a | 1 | **87.3** | **30.7** | 10.3 | 29.8 | 9.9 |
| 50S | 2120 | n/a | 1 | **77.7** | **21.4** | 7.7 | 27.8 | 9.3 |
| | 2121 | n/a | 1 | **104.2** | **30.7** | 10.2 | 30 | 10 |
| | 2122 | n/a | 1 | **130.7** | **30.7** | 11.5 | 23.8 | 8.9 |
| 53S | 2154 | n/a | 1 | **162.7** | **30.7** | 13.3 | 20.3 | 6.8 |
| | 2155 | n/a | 2 | **278.8** | **30.7** | 10.2 | 13.5 | 4.5 |
| | 2156 | n/a | 4 | **459.3** | **30.7** | 10.2 | 9 | 3 |
| | 2157 | n/a | 4 | **509.9** | **30.7** | 10.4 | 7.7 | 2.6 |

## C.16 AS/400e Custom Application Server Model SB1

AS/400e application servers are particularly suited for environments with minimal database needs, minimal disk storage needs, lots of low-cost memory, high-speed connectivity to a database server, and minimal upgrade importance.

The throughput rates for Financial (FI) dialogsteps (ds) per hour may be used to size systems for customer orders. **Note: 1 SD ds = = 2.5 Fl ds**. (SD = Sales & Distribution).

| *Table C.16.1 AS/400e Custom Application Server Model SB1* | | | | |
|---|---|---|---|---|
| **Model** | **CPUs** | **SAP Release** | **SD ds/hr @ 65% CPU Utilization** | **FI ds/hr @ 65% CPU Utilization** |
| 2312 | 8 | 3.1H | 109,770.49 | 274,426.23 |
| | | 4.0B | 65,862.29 | 164,655.74 |
| 2313 | 12 | 3.1H | 158,715.76 | 396,789.40 |
| | | 4.0B | 95,229.46 | 238,073.64 |

## C.17 AS/400 Models 4xx, 5xx and 6xx Systems

| *Table C.17.1 AS/400 RISC Systems* | | | | | | |
|---|---|---|---|---|---|---|
| **Model** | **Feature Code** | **CPUs** | **Memory (MB) Maximum** | **Disk (GB) Maximum** | **V3R7 CPW** | **V4R1 CPW** |
| 400 | 2130 | 1 | 160 | 50 | 13.8 | 13.8 |
| | 2131 | 1 | 224 | 50 | 20.6 | 20.6 |
| | 2132 | 1 | 224 | 50 | 27 | 27 |
| | 2133 | 1 | 224 | 50 | 33.3 | 35 |
| 500 | 2140 | 1 | 768 | 652 | 21.4 | 21.4 |
| | 2141 | 1 | 768 | 652 | 30.7 | 30.7 |
| | 2142 | 1 | 1024 | 652 | 43.9 | 43.9 |
| 510 | 2143 | 1 | 1024 | 652 | 77.7 | 81.6 |
| | 2144 | 1 | 1024 | 652 | 104.2 | 111.5 |
| 530 | 2150 | 1 | 4096 | 996 | 131.1 | 148 |
| | 2151 | 1 | 4096 | 996 | 162.7 | 188.2 |
| | 2152 | 2 | 4096 | 996 | 278.8 | 319 |
| | 2153 | 4 | 4096 | 996 | 459.3 | 598 |
| | 2162 | 4 | 4096 | 996 | 509.9 | 650 |

| *Table C.17.2 AS/400e Systems* | | | | | |
|---|---|---|---|---|---|
| **Model** | **Feature Code** | **CPUs** | **Memory (MB) Maximum** | **Disk (GB) Maximum** | **V4R3 CPW** |
| 600 | 2129 | 1 | 384 | 175.4 | 22.7 |
| | 2134 | 1 | 384 | 175.4 | 32.5 |
| | 2135 | 1 | 384 | 175.4 | 45.4 |
| | 2136 | 1 | 512 | 175.4 | 73.1 |
| 620 | 2175 | 1 | 1856 | 944.8 | 50 |
| | 2179 | 1 | 2048 | 944.8 | 85.6 |
| | 2180 | 1 | 2048 | 944.8 | 113.8 |
| | 2181 | 1 | 2048 | 944.8 | 210 |
| | 2182 | 2 | 4096 | 944.8 | 464.3 |
| 640 | 2237 | 1 | 16384 | 1340 | 319 |
| | 2238 | 2 | 8704 | 1340 | 583.3 |
| | 2239 | 4 | 16384 | 1340 | 998.6 |
| 650 | 2188 | 8 | 40960 | 2095.9 | 3660 |
| | 2189 | 12 | 40960 | 2095.9 | 4550 |
| | 2240 | 8 | 32768 | 2095.9 | 1794 |
| | 2243 | 12 | 32768 | 2095.9 | 2340 |

## C.18 AS/400 CISC Model Capacities

| *Table C.18.1 AS/400 CISC Model: 9401* | | | | | |
|---|---|---|---|---|---|
| **Model** | **Feature** | **CPUs** | **Memory (MB) Maximum** | **Disk (GB) Maximum** | **CPW** |
| P02 | n/a | 1 | 16 | 2.1 | 7.3 |
| P03 | 2114 | 1 | 24 | 2.99 | 7.3 |
| | 2115 | 1 | 40 | 3.93 | 9.6 |
| | 2117 | 1 | 56 | 3.93 | 16.8 |

| *Table C.18.2 AS/400 CISC Model: 9402 Systems* | | | | |
|---|---|---|---|---|
| **Model** | **CPUs** | **Memory (MB) Maximum** | **Disk (GB) Maximum** | **CPW** |
| C04 | 1 | 12 | 1.3 | 3.1 |
| C06 | 1 | 16 | 1.3 | 3.6 |
| D02 | 1 | 16 | 1.2 | 3.8 |
| D04 | 1 | 16 | 1.6 | 4.4 |
| E02 | 1 | 24 | 2.0 | 4.5 |
| D06 | 1 | 20 | 1.6 | 5.5 |
| E04 | 1 | 24 | 4.0 | 5.5 |
| F02 | 1 | 24 | 2.1 | 5.5 |
| F04 | 1 | 24 | 4.1 | 7.3 |
| E06 | 1 | 40 | 7.9 | 7.3 |
| F06 | 1 | 40 | 8.2 | 9.6 |

| *Table C.18.3 AS/400 CISC Model: 9402 Servers* | | | | | |
|---|---|---|---|---|---|
| **Feature Code** | **CPUs** | **Memory (MB) Maximum** | **Disk (GB) Maximum** | **C/S CPW** | **Interactive CPW** |
| S01 | 1 | 56 | 3.9 | 17.1 | 5.5 |
| 100 | 1 | 56 | 7.9 | 17.1 | 5.5 |

| *Table C.18.4 AS/400 CISC Model: 9404 Systems* | | | | |
|---|---|---|---|---|
| **Model** | **CPUs** | **Memory (MB) Maximum** | **Disk (GB) Maximum** | **CPW** |
| B10 | 1 | 16 | 1.9 | 2.9 |
| C10 | 1 | 20 | 1.9 | 3.9 |
| B20 | 1 | 28 | 3.8 | 5.1 |
| C20 | 1 | 32 | 3.8 | 5.3 |
| D10 | 1 | 32 | 4.8 | 5.3 |
| C25 | 1 | 40 | 3.8 | 6.1 |
| D20 | 1 | 40 | 4.8 | 6.8 |
| E10 | 1 | 40 | 19.7 | 7.6 |
| D25 | 1 | 64 | 6.4 | 9.7 |
| F10 | 1 | 72 | 20.6 | 9.6 |
| E20 | 1 | 72 | 19.7 | 9.7 |
| F20 | 1 | 80 | 20.6 | 11.6 |
| E25 | 1 | 80 | 19.7 | 11.8 |
| F25 | 1 | 80 | 20.6 | 13.7 |

| *Table C.18.5 AS/400 CISC Model: 9404 Servers* | | | | | |
|---|---|---|---|---|---|
| **Feature Code** | **CPUs** | **Memory (MB) Maximum** | **Disk (GB) Maximum** | **C/S CPW** | **Interactive CPW** |
| 135 | 1 | 384 | 27.5 | 32.3 | 9.6 |
| 140 | 2 | 512 | 47.2 | 65.6 | 11.6 |

| *Table C.18.6 AS/400 CISC Model: 9406 Systems* | | | | |
|---|---|---|---|---|
| **Model** | **CPUs** | **Memory (MB) Maximum** | **Disk (GB) Maximum** | **CPW** |
| B30 | 1 | 36 | 13.7 | 3.8 |
| B35 | 1 | 40 | 13.7 | 4.6 |
| B40 | 1 | 40 | 13.7 | 5.2 |
| B45 | 1 | 40 | 13.7 | 6.5 |
| D35 | 1 | 72 | 67.0 | 7.4 |
| B50 | 1 | 48 | 27.4 | 9.3 |
| E35 | 1 | 72 | 67.0 | 9.7 |
| D45 | 1 | 80 | 67.0 | 10.8 |
| D50 | 1 | 128 | 98.0 | 13.3 |
| E45 | 1 | 80 | 67.0 | 13.8 |
| F35 | 1 | 80 | 67.0 | 13.7 |
| B60 | 1 | 96 | 54.8 | 15.1 |
| F45 | 1 | 80 | 67.0 | 17.1 |
| E50 | 1 | 128 | 98.0 | 18.1 |
| B70 | 1 | 192 | 54.8 | 20.0 |
| D60 | 1 | 192 | 146 | 23.9 |
| F50 | 1 | 192 | 114 | 27.8 |
| E60 | 1 | 192 | 146 | 28.1 |
| D70 | 1 | 256 | 146 | 32.3 |
| E70 | 1 | 256 | 146 | 39.2 |
| F60 | 1 | 384 | 146 | 40.0 |
| D80 | 2 | 384 | 256 | 56.6 |
| F70 | 1 | 512 | 256 | 57.0 |
| E80 | 2 | 512 | 256 | 69.4 |
| E90 | 3 | 1024 | 256 | 96.7 |
| F80 | 2 | 768 | 256 | 97.1 |
| E95 | 4 | 1152 | 256 | 116.6 |
| F90 | 3 | 1024 | 256 | 127.7 |
| F95 | 4 | 1280 | 256 | 148.8 |
| F97 | 4 | 1536 | 256 | 177.4 |

| *Table C.18.7 AS/400 Advanced Systems (CISC)* | | | | | |
|---|---|---|---|---|---|
| **Model** | **Feature Code** | **CPUs** | **Memory (MB) Maximum** | **Disk (GB) Maximum** | **CPW** |
| 200 | 2030 | 1 | 24 | 23.6 | 7.3 |
| | 2031 | 1 | 56 | 23.6 | 11.6 |
| | 2032 | 1 | 128 | 23.6 | 16.8 |
| 300 | 2040 | 1 | 72 | 117.4 | 11.6 |
| | 2041 | 1 | 80 | 117.4 | 16.8 |
| | 2042 | 1 | 160 | 117.4 | 21.1 |
| 310 | 2043 | 1 | 832 | 159.3 | 33.8 |
| | 2044 | 2 | 832 | 159.3 | 56.5 |
| 320 | 2050 | 1 | 1536 | 259.6 | 67.5 |
| | 2051 | 2 | 1536 | 259.6 | 120.3 |
| | 2052 | 4 | 1536 | 259.6 | 177.4 |

| *Table C.18.8 AS/400 Advanced Servers (CISC)* | | | | | | |
|---|---|---|---|---|---|---|
| **Model** | **Feature Code** | **CPUs** | **Memory (MB) Maximum** | **Disk (GB) Maximum** | **C/S CPW** | **Interactive CPW** |
| 20S | 2010 | 1 | 128 | 23.6 | 17.1 | 5.5 |
| 2FS | 2010 | 1 | 128 | 7.8 | 17.1 | 5.5 |
| 2SG | 2010 | 1 | 128 | 7.8 | 17.1 | 5.5 |
| 2SS | 2010 | 1 | 128 | 7.8 | 17.1 | 5.5 |
| 30S | 2411 | 1 | 384 | 86.5 | 32.3 | 9.6 |
| | 2412 | 2 | 832 | 86.5 | 68.5 | 11.6 |